

API of the Catalog of Marked Goods

Version 5.38

Contents

What's new in v.5.38 dd. 03/03/2025.....	4
1. GENERAL INFORMATION ABOUT THE CATALOG OF MARKED GOODS.....	5
2. GENERAL PARAMETERS FOR WORKING WITH THE NC API.....	6
2.1. Limit of API Requests.....	6
2.2. Universal Request Parameters.....	6
2.3. Allowed Unicode characters when creating and updating the goods cards.....	7
2.4. HTTP ETag (Version Control).....	7
2.5. Response Codes.....	8
3. METHODS TO WORK WITH API of NC.....	10
3.1. Getting information about the card.....	10
3.1.1. Method "Get information about your own card".....	10
3.1.2. Method "Get information about published card".....	22
3.1.3. Method "Get brief information about published card".....	32
3.1.4. Method "Get a list of your own cards with brief information about them".....	44
3.1.5. Method "Check changes in cards".....	47
3.1.6. Method "Check whether cards or FEACN codes belong to marked goods groups".....	49
3.2. Create or edit a card.....	53
3.2.1. Method "Generate a goods code".....	53
3.2.2. Method "Create or edit a card".....	55
3.2.3. Method "Check update package processing status".....	67
3.2.4. Method "Change the photo size".....	74
3.2.5. Method "Forcibly send a card for moderation".....	75
3.3. Getting information about attributes.....	76
3.3.1. Method "Get a tree of categories".....	76
3.3.2. Method "Get an attribute list".....	79
3.3.3. Method "Get a directory of countries of manufacture".....	84
3.3.4. Method "Get a catalog of trademarks".....	86
3.3.5. Method "Check the presence of permit document in the directory".....	88
3.3.6. Method "Get information about a permitting document by a goods code and INN".....	90
3.3.7. Method "Get information about a permitting document by a number and date".....	94
3.4. Signing a card.....	98

3.4.1. Method “Get XML for subsequent signing a card”	98
3.4.2. (Deprecated) Method “Sign a card by using an attached signature”	100
3.4.3. Method “Sign a card by using the detached signature”	101
3.5. Working with subaccounts.....	103
3.5.1. Method “Get a list of company subaccounts”	103
3.5.2. Method “Get a list of companies and goods codes for which access has been granted, by subaccount”	105
3.5.3. Method “Get XML file that is required to control access to subaccounts”	107
3.5.4. Method “Sign granting or revoking permission to a subaccount to use goods codes for MCs emission”	110
4. CATALOGS.....	114
4.1. Catalog – List of supported goods groups.....	114
5. Modifications introduced in the previous versions of the document.....	116

What's new in v.5.38 dd. 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](#)”.

To view a full document revision history, go to the “[5. Modifications introduced in the previous versions of the document](#)” section.

1. 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.

2. 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> — test environment address;
- [api.национальный-каталог.рф](https://api.natsionalnyy-katalog.rf) — 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](#)”.

2.1. 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.

2.2. 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"

2.3. 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	

2.4. 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

2.5. 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.

3. METHODS TO WORK WITH API of NC

3.1. Getting information about the card

3.1.1. 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

Parameter	Type	Mandatory	Description	Comment
				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
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”;

Parameter	Type	Mandator y	Description	Comment
				“show-pack” — “Show-pack”
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” — 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

Parameter	Type	Mandatory	Description	Comment
flags_updated_date	string	+	Date of flags update	
create_date	string	+	Card creation date	
update_date	string	+	Date of card update	
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" — e-commerce photo; "undef" — single shot, goods item photo from

Parameter	Type	Mandatory	Description	Comment
				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	
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

Parameter	Type	Mandatory	Description	Comment
*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
*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":"00000000000001",
          "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": "00000000000001",
        "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": "00000000000001",
        "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": "00000000000001",
        "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": "00000000000001",
        "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": "00000000000001",
        "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":"00000000000001",
            "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>00000000000002</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>ПОСТ-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>

```

3.1.2. 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>in=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
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	Default value is 1

Parameter	Type	Mandatory	Description	Comment
			packing	
*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	
*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	

Parameter	Type	Mandatory	Description	Comment
*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
*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;

Parameter	Type	Mandator y	Description	Comment
				“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; “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.пф/product/00000000000002-ru-nabor-odekolon-100-ml-shampun-dlya-volos-i-tela-75-ml-parfyumirovanny-dezodorant-dlya-tela-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 парфюмерно-  
косметической 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": "КОРОВКА/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": "00000000000002",
      "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": "00000000000002",
        "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/00000000000002-ru-nabor-
odekolon-100-ml-shampun-dlya-volos-i-tela-75-ml-parfyumirovanny-dezodorant-dlya-tela-
50-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 парфюмерно-
косметической 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>КОРОВКА/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>

```

3.1.3. 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

Parameter	Type	Mandator y	Description	Comment
				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	
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:

Parameter	Type	Mandatory	Description	Comment
				“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	
*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	

Parameter	Type	Mandator y	Description	Comment
*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” 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;

Parameter	Type	Mandatory	Description	Comment
				“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": "00000000000001",
          "type": "ntin",
          "multiplier": 1,
          "level": "trade-unit"
        },
        {
          "value": "00000000000001",
          "type": "gtin",
          "multiplier": 1,
          "level": "trade-unit"
        },
        {
          "value": "00000000000002",
          "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>00000000000001</value>
          <type>gtin</type>
          <multiplier>1</multiplier>
          <level>trade-unit</level>
        </item>
        <item>
          <value>00000000000002</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>00000000000001</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": "00000000000001",
          "type": "ntin",
          "multiplier": 1,
          "level": "trade-unit"
        },
        {
          "value": "00000000000001",
          "type": "gtin",
          "multiplier": 1,
          "level": "trade-unit"
        },
        {
          "value": "00000000000002",
          "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>00000000000001</value>
          <type>gtin</type>
          <multiplier>1</multiplier>
          <level>trade-unit</level>
        </item>
        <item>
          <value>00000000000002</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>

```

3.1.4. 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 YYYY-MM-DD HH:ii:ss	All “gtin” updated within a month before the specified date will be selected
from_date	string	-	Date and time in format YYYY-	All “gtin” updated within a month after the specified

Parameter	Type	Mandatory	Description	Comment
			MM-DD HH:ii:ss	date will be selected
good_statuses	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_statuses	array	-	Array of the current statuses of the goods card	Possible values: “draft” — draft; “moderation” — being

Parameter	Type	Mandatory	Description	Comment
				moderated; “errors” — to be changed; “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": "00000000000001",
        "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>00000000000001</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>
```

```
        </item>
    </goods>
</result>
</root>
```

3.1.5. 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	
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>

```

3.1.6. 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":[
    "0000000000000001",
    "0000000000000002",
    "0000000000000003"
  ],
  "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

Parameter	Type	Mandatory	Description	Comment
				“029”): “0” — “Remaining items not to be marked” “1” — “Remaining items to be marked” “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":"00000000000001",
        "tnved":"3303",
        "is_marked":"Товар подлежит маркировке",
        "is_marked_code":1
      },
      {
        "gtin":"00000000000002",
        "tnved":null,
        "is_marked":"Товар не найден",
        "is_marked_code":2
      },
      {
        "gtin":"00000000000003",
        "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>0000000000001</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>

```

```
</tnveds>
</result>
</root>
```

3.2. Create or edit a card

3.2.1. 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 authentication token	This parameter is mandatory if “apikey” (“Identifier (key) of the goods owner”) is 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": "0000000000001"
      },
      {
        "gtin": "0000000000002"
      },
      {
        "gtin": "0000000000003"
      }
    ]
  }
}
```

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>0000000000001</gtin>
  <gtin>0000000000002</gtin>
  <gtin>0000000000003</gtin>
</drafts>
</result>
</root>

```

3.2.2. 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	It is designated for laboratories and other suppliers of content that

Parameter	Type	Mandatory	Description	Comment
y	g		manufacturer of goods	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	It is mandatory when creating a goods card with technical goods code (in this case the goods code is not specified in

Parameter	Type	Mandatory	Description	Comment
			with technical code of the goods	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
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 “mark-check”
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: <ul style="list-style-type: none"> 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

Parameter	Type	Mandator y	Description	Comment
				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	<p>Possible values:</p> <p>“trade-unit” — consumer; “inner-pack” — group consumer; “box” — shipping.</p> <p>For all types of packages, except “trade-unit”, it is recommended to indicate explicitly in the attribute array:</p> <p>“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.</p> <p>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:</p> <p>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)</p>
*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

Parameter	Type	Mandatory	Description	Comment
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	<p>Optional when editing and deleting.</p> <p>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 manufacture” attribute).</p> <p>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 “YYYY-MM-DD”, for the attributes that have “attr_id” (“Attribute identifier”) equal to:</p> <p>“23555” (“Marketing authorization”); “23890” (“Marketing authorization in the State Register”); “23561” (“Certificate of conformity”); “23557” (“Declaration of conformity”).</p> <p>And there is no need to put spaces before and after the delimiter “:::”</p>
*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	<p>It is available only when editing the existing goods.</p> <p>Possible values:</p> <p>“1” – deletion of the goods attribute.</p> <p>When sending this parameter, “attr_value” shall be specified with a deleted value</p>
good_images	array	-	Array of images	
*photo_type	string	+	Image type	<p>Possible values:</p> <p>"default" — default photo (front view); "7" — left-side photo of goods; "19" — right-side photo of goods; "13" — back-side photo of goods;</p>

Parameter	Type	Mandatory	Description	Comment
				"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 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 <TOKEH>"
--data-raw "{
  "gtin":"0000000000001",
  "tnved":"3303",
  "brand":"string",
  "categories":[
    30120
  ],
  "good_name":"Полное наименование товара",
  "identified_by":[
    {
      "value":"0000000000001",
      "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 <TOKEH>"
--data-raw "{
    \"gtin\": \"00000000000001\",
    \"tnved\": \"3303\",
    \"brand\": \"string\",
    \"categories\": [
        30570
    ],
    \"good_name\": \"Полное наименование товара\",
    \"identified_by\": [
        {
            \"value\": \"00000000000001\",
            \"type\": \"gtin\",
            \"multiplier\": 1,
            \"level\": \"trade-unit\",
            \"unit\": \"шт\"
        },
        {
            \"value\": \"000000000000002\",
            \"type\": \"gtin\",
            \"multiplier\": 10,
            \"level\": \"inner-pack\",
            \"unit\": \"шт\"
        },
        {
            \"value\": \"000000000000003\",
            \"type\": \"gtin\",
            \"multiplier\": 100,
            \"level\": \"box\",
            \"unit\": \"шт\"
        }
    ],
    \"good_attrs\": [
        {
            \"attr_id\": 13763,
            \"attr_value_type\": \"gtin\",
            \"attr_value\": \"00000000000001\",
            \"gtin\": \"000000000000002\"
        },
        {
            \"attr_id\": 13763,
            \"attr_value_type\": \"gtin\",
            \"attr_value\": \"000000000000002\",
            \"gtin\": \"000000000000003\"
        }
    ]
}"
```

```

    },
    {
      "attr_id":4424,
      "attr_value ":"Наименование упаковки товара",
      "gtin":"000000000000002"
    },
    {
      "attr_id":2439,
      "attr_value":"40",
      "gtin":"000000000000002"
    },
    {
      "attr_id":2437,
      "attr_value":"15",
      "gtin":"000000000000002"
    },
    {
      "attr_id":2438,
      "attr_value":"40",
      "gtin":"000000000000002"
    },
    {
      "attr_id":2710,
      "attr_value":"КОРОВКА/БОКС",
      "gtin":"000000000000002"
    },
    {
      "attr_id":2713,
      "attr_value":"КАПТОН",
      "gtin":"000000000000002"
    }
  ]
}"

```

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 <TOKEN>"
--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\":\"0000000000000001\",
  \"good_name\":\"Шоколад с Цельным Миндалём 55% 90г 14шт шоу-бокс\",
  \"identified_by\":[
    {
      \"value\":\"0000000000000001\",
      \"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\":\"0000000000000001\",
      \"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 <TOKEH>"
--data-raw "
    <?xml version="1.0" encoding="UTF-8"?>
    <entries>
        <entry>
            <gtin>000000000000001</gtin>
            <tnved>0403</tnved>
            <good_name>Йогурт легкий. Злаки + чернослив</good_name>
            <identified_by>
                <item>
                    <value>000000000000001</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>000000000000001</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 <TOKEH>"
--data-raw "{
  \"gtin\":\"00000000000004\",
  \"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 <TOKEH>"
--data-raw "
  <?xml version=\"1.0\" encoding=\"UTF-8\"?>
  <entries>
    <entry>
      <gtin>00000000000004</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 <TOKEH>"
--data-raw "{
  \"good_id\":1939447,
  \"identified_by\":[
    {
      \"value\":\"00000000000001\",
      \"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 <TOKEH>"
--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>π</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>
```

3.2.3. 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/feed-status?
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 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	

Parameter	Type	Mandatory	Description	Comment
***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	

Parameter	Type	Mandator y	Description	Comment
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	
*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": "00000000000001",
        "good_id": "789817",
        "attribute_id": "2716",
        "attribute_name": "Заявленный объем",
        "status_code": 5,
        "status_message": "Отменено",
        "message": "Неверный объем"
      },
      {
        "id": 1,
        "gtin": "00000000000001",
        "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>00000000000001</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>00000000000001</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
Единom 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>

```

3.2.4. 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://национальный-каталог.рф/i/
300x200/5a7eb614-13d3-69ed-caf7-420624d1bdd3.jpg&width=300&height=400

```

Request string parameters:

Parameter	Type	Mandator y	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)
no-background	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

Parameter	Type	Mandatory	Description	Comment
				“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.

3.2.5. 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/feed-moderation?apikey=XXX&gtin=000000000000001&inn=12345678
```

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

Parameter	Type	Mandatory	Description	Comment
	g			

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 02XXXXXXXXX601050000383 не в том статусе."
  }
}
```

3.3. Getting information about attributes

3.3.1. 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
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>
```

3.3.2. 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 chestnyyznak.rf.

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 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”	It is used to receive a set of attributes

Parameter	Type	Mandatory	Description	Comment
			type	
attr_type	string	-	Attribute type	<p>If “attr_type” parameter is specified, then either “cat_id”, or “tnved”, or “is_set” shall be indicated.</p> <p>Available values:</p> <p>"a" — (default value) return all attributes; "m" — return mandatory attributes only; "r" — return recommended attributes only; "o" — return optional attributes only</p>

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	<p>Possible values:</p> <p>“true” — yes; “false” — no</p>
attr_multiplicity	boolean	+	Attribute of multiplicity attribute	<p>Possible values:</p> <p>“true” — yes; “false” — no.</p> <p>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” attribute.</p>
attr_multiplicity_type	string	+	Type of attribute multiplicity	<p>Possible values:</p> <p>“regular” is a multiplicity attribute without a check of the attribute type for uniqueness</p>

Parameter	Type	Mandator y	Description	Comment
				(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 to order the marking code)	It accepts values true/false
second_layer	boolean	+	Indicator that the attribute belongs to the second layer of	It accepts values true/false

Parameter	Type	Mandatory	Description	Comment
			attributes	
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>
  </result>
</root>

```

```

<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>

```

3.3.3. 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

Parameter	Type	Mandatory	Description	Comment
	g		owner	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

Response parameters:

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

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

```
{
  "apiversion": 3,
  "result": {
    "_etag": "0a23f98d522e7c05",
    "_list": [
      {
        "country_iso": "AD",
        "country_name": "Андорра"
      },
      {
        "country_iso": "AE",
        "country_name": "ОАЭ"
      },
      {
        "country_iso": "AF",
        "country_name": "Афганистан"
      },
      {
        "country_iso": "AG",
        "country_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>
        <country_iso>AD</country_iso>
        <country_name>Андорра</country_name>
      </item>
      <item>
        <country_iso>AE</country_iso>
        <country_name>ОАЭ</country_name>
      </item>
      <item>
        <country_iso>AF</country_iso>
        <country_name>Афганистан</country_name>
      </item>
      <item>
        <country_iso>AG</country_iso>
        <country_name>Антигуа и Барбуда</country_name>
      </item>
    </_list>
  </result>
</root>
```

3.3.4. 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	Minimum allowed value is 0

Parameter	Type	Mandatory	Description	Comment
	string		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	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>

```

3.3.5. 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	Mandator y	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 registration of the permit document	It is used only with the full number of the 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": "БИИТ"
}"
```

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.PA01.B.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	
*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"
        }
      ]
    }
  ]
}
```

3.3.6. 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 Russian permitting documents that 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\": \"000000000000001\",
  \"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;

Parameter	Type	Mandatory	Description	Comment
				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; 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_inved	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:

Parameter	Type	Mandator y	Description	Comment
				declaration of conformity; certificate of conformity
*active	boolean	-	Attribute of certificate activity	Only for the state registration certificate
*update_date	datetim e	-	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":"EA3C N RU Д-UZ.0000.B.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"
    }
}
]
}
}

```

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

The method “v4/rd-info” allows to obtain full information about Russian permitting documents (Declaration of conformity, Certificate of conformity, State registration certificate):

- 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).

If information about EEU permitting documents is requested, an error will be returned.

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.E.005115.11.17\"
},
{
  \"attr_id\": 23557,
  \"number\": \"EA9C N RU Д-UZ.PA07.B.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 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

Parameter	Type	Mandatory	Description	Comment
*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_inved	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;

Parameter	Type	Mandator y	Description	Comment
				certificate of conformity
*active	boolean	-	Attribute of certificate activity	Only for the state registration certificate
*update_date	datetim e	-	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"
    }
}
}
}

```

3.4. Signing a card

3.4.1. 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 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\": [
    \"0000000000000001\",
    \"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 скомпрометирована.
        Попробуйте еще раз."
      },
      {
        "GTIN": "123123123123",
        "message": "Не удалось получить товар по GTIN"
      }
    ]
  }
}

```

3.4.2. (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-product-document](#). 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	This parameter is mandatory if “apikey” (“Identifier (key) of the goods owner”) is

Parameter	Type	Mandatory	Description	Comment
			authentication token	not specified

Request body parameters:

Parameter	Type	Mandatory	Description	Comment
goodId	integer	+	Identifier of a goods item for which XML is transmitted	
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 скомпрометирована.
Попробуйте еще раз."
        }
      ]
    }
  }
}

```

3.4.3. 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":
"PD94bWwgdmVyc2lvdj1cIjEuMFwiIGVuY29kaW5nPVwiVVRGLThcIj8+XG48Z29vZD4=...",
    "signature": "..."
  },
  {
    "goodId": 3119690,
    "base64Xml":
"PD94bWwgdmVyc2lvdj1cIjEuMFwiIGVuY29kaW5nPVwiVVRGLThcIj8+XG48Z29vZD4=...",
    "signature": "..."
  },
  {
    "goodId": 521,
    "base64Xml":
"PD94bWwgdmVyc2lvdj1cIjEuMFwiIGVuY29kaW5nPVwiVVRGLThcIj8+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 feed-product-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"
    }
  ]
}
}

```

3.5. Working with subaccounts

3.5.1. 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	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
linked_accounts	array	+	Array of company sub-accounts	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 limited access	Possible values: “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":"00000000000001"
          },
          {
            "gtin":"00000000000002"
          }
        ]
      }
    ]
  }
}
```

```

    }
}

```

3.5.2. 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":"00000000000001"
      },
      {
        "producer_inn":"inn2",
        "producer_name":"ООО Василек",
        "gtin":"00000000000002"
      }
    ]
  }
}
```

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
    }
]
}
}

```

3.5.3. 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

Parameter	Type	Mandatory	Description	Comment
				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
      }
    ]
  }
}
```

3.5.4. 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	
signature	string	+	Detached signature (in base64 format) corresponding to CAdES type in pkcs7 format	

Request body example:

```
[
  {
    "linked_account_inn": "inn1",

    "base64_xml": "PD94bWwgdmVyc2lvdj1cIjEuMFwiIGVuY29kaW5nPVwiVVRGLThcIj8+XG48YWNjb3VudD48c
2NoZWlhVmVyc2lvdj42LjA8L3NjaGVtYVZlcnNpb24+PG1haW5BY2NvdW50SW5uPg==...",
    "signature": "..."
  },
  {
    "linked_account_inn": "inn2",

    "base64_xml": "PD94bWwgdmVyc2lvdj1cIjEuMFwiIGVuY29kaW5nPVwiVVRGLThcIj8+XG48YWNjb3VudD48c
2NoZWlhVmVyc2lvdj42LjA8L3NjaGVtYVZlcnNpb24+PG1haW5BY2NvdW50SW5uPg==...",
    "signature": "..."
  },
  {
    "linked_account_inn": "inn3",
```

```

"base64_xml": "PD94bWwgdmVyc2lvdj1cIjEuMFwiIGVuY29kaW5nPVwiVVRGLThcIj8+XG48YWNjb3VudD48c
2NoZW1hVmVyc2lvdj42LjA8L3NjaGVtYVZlc3Npb24+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
    }
  ]
}
```

4. CATALOGS

4.1. 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	nep	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

5. Modifications introduced in the previous versions of the document

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>.

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 certificate 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/>. 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](#), [short-product](#). 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 [linked-accounts](#) 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 [linked-accounts-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	5.38
---------	------

Last updated 2025-03-03 11:43:46 UTC