DEPARTMENT OF THE STATE SIGNS
OF THE MINISTRY OF FINANCE OF THE REPUBLIC OF BELARUS

Republican Unitary Enterprise
"Belblankavyd Publishing House"

STATE INFORMATION SYSTEM FOR MARKING GOODS WITH UNIFIED CONTROL MARKS OR IDENTIFICATION MEANS (SIS "ELECTRONIC MARK")

WEB SERVICE OF CROSS-SYSTEM COUPLING

API specification v.3.8.1

Merchandise marking and turnover

Revision 3.20

Minsk
2023

RECORD OF CHANGES

Revision	Date	Change description	Author(s)
1	2	3	4
3.0	14.10.2021	Revision of the document describing API v.3.0.0. A new code type has been added: `20` (“Footwear code Belarus”). Code types `5`, `14` and `15` have been excluded from the available for ordering by methods `/v2/orders/add` and `/v2/orders/addGroupOrders`. New leftovers labelling functionality has been added: - new product group `leftovers_shoes` (“Leftovers. Footwear”) has been added; - catalogue section `10092` (“Leftovers. Footwear products”) has been added; - unified GTIN `04814130000030` for ordering code labels for footwear leftovers has been introduced; - leftovers inventory output submission method `/v2/items/inventory` has been added; - method to obtain the list of inventory sheets `/v2/items/inventoryList` has been added; - leftovers labelling report submission method `/v2/reports/addMarkLeftovers` has been added; - new status (“Introduced into turnover in Belarus”) has been added for label code `55`. New methods have been added: - `/labels/changeOwner` to change the current label code owner: - `/v2/reports/addMarkCirculation` for a product introduction into turnover report. Examples of `/v2/reports` method return have been expanded. New report types have been added: - type `6` (“Introduction into Turnover. Leftovers”); - type `7` (“Introduction into Turnover in Belarus”). Method `/ships/add` и объект `Ship` has been extended with optional set of parameters `eas_products` for exchange of additional data on goods price, VAT amount and documentary recognition of conformity to quality and/or safety regulations in frames of transborder trade between the EAEU member states. New enumerations have been added: - `circulation_reason` (“Reasons to Introduce	Valery Matokha

Revision	Date	Change description	Author(s)
1	2	3	4
		into Turnover"); - `certificate_types` ("Types of Certificates"). Appendox A has been updated with introduced amendments and supplements. Appendix C has been revised.	
3.1	03.11.2021	Revision of the document describing API v.3.0.1. Item 2.6 about screening of special command delimiting characters ASCII 29 has been added. Additional parameters `org_number` and `org_name` have been added to turnover method on report introduction `/v2/reports/addMarkCirculation`. A note about application of new format serial numbers, which contain EAEU member state numerical ID in the first digit to indicate the member state the label is issued in, is added to Appendix C, C.1.2.	Valery Matokha
3.2	10.11.2021	Revision of the document describing API v.3.0.2. Description of optional parameter `is_import` (search attribute in the imported goods catalogue) and return options has been added to goods search method `/items/findItems`. Limitations of label code transmission have been introduced to the following methods: - goods introduction into turnover report transmission method `/v2/reports/addMarkCirculation`; - leftovers labelling report transmission method `/v2/reports/addMarkLeftovers`. Label code `20` has been renamed as ("Goods Code Belarus (Footwear, Tyres and Wheel Coverings"). Leftovers labelling functionality has been extended: - new product group `leftovers_tires` ("Leftovers. Tyres and Wheel Coverings") has been added; - catalogue section `10093` ("Leftovers. Tyres and Wheel Coverings") has been added; - unified GTIN `04814130000047` for ordering code labels for tyre leftovers has been introduced. Packaged water labelling functionality has been extended: - new product group `water` ("Packaged water") has been added;	Valery Matokha

Revision	Date	Change description	Author(s)
1	2	3	4
		- catalogue section `10094` (“Packaged water”) has been added. In Appendix C: - description for sections C.1, C.2 and C.4 have been added; - new section C.5 has been added.	
3.3	16.11.2021	Amended and complemented revision of the document describing API v.3.0.2. Optional input parameters for the following methods have been specified: - code cancellation `/labels/destroy`; - list of cancellations `/stats/destroyList`. A limit is added to `operation_date` parameter description in the code shipment method `/ships/add`.	Valery Matokha
3.4	02.12.2021	Revision of the document describing API v.3.1.0. A report on the previously used UCMs can be corrected: - a new method to completely or partially cancel UCM application is added: `/v2/reports/addCancelUsed` - a new report type is added: `8 – Cancel Application` Dairy products shipment recording method (`/ships/gtin`) and `Ship` item are supplemented with optional parameters for transmission of additional data on the goods price, VAT amount and documentary proof of goods qualification as to quality and/or safety within transborder trade between the EAEU member states. Trusted acceptance method for footwear and tyre shipments from Russia is added (temporary mode to be modified in the following API versions): - List of shipments: `/ships/importList` - Acceptance of shipments: `/ships/addImport`; - Shipment information: `/ships/import`. Description of: - Appendix B, - Appendix C, Table C.2.1, - Appendix C, C.3 has been revised.	Valery Matokha
3.5	04.01.2022	Revision of the document describing API v.3.1.1. Leftovers labelling functionality has been extended:	Valery Matokha

Revision	Date	Change description	Author(s)
1	2	3	4
		- product group `leftovers_clothes` (“Leftovers. Consumer goods”) has been added; - catalogue section `10095` has been added (“Leftovers. Consumer goods”); - unified GTIN `04814130000054` has been introduced to order consumer goods leftovers marking codes. Sections C.1, C.2 and C.4 in Appendix C have been updated. Table C.2.1 has been extracted into separate Appendix D.	
3.6	25.01.2022	Revision of the document describing API v.3.2.0. Restrictions on the number of ordered and shipped marking codes have been clarified. The description of the `/items/edit` method for changing product information, used to add (change) an image of a product has been updated. Additional parameter `document_id` have been added to turnover method on report introduction `/v2/reports/addMarkCirculation`. Examples of response options returned by the `/v2/reports` method have been extended. The description of the structure of delivery information details has been updated. An object of the `Import` class (see section 4.1.12). Appendix B has updated descriptions of Sections B.5. Table D.1 of Appendix D has been supplemented with a new labelling regime “Import from the EAEU Member States”. Misprints and inaccuracies have been corrected in the text of the document.	Valery Matokha
3.7	12.03.2022	Revision of the document describing API v.3.2.1. Additional parameter `document_id` have been added to dairy products shipment recording method `/ships/gtin` and the requirements for some attributes have been changed. Table D.1 of Appendix D for the product group “Dairy products” expands the list of abelling purposes and the composition corresponds to the abelling parameters.	Valery Matokha
3.8	30.03.2022	Revision of the document describing	Valery

Revision	Date	Change description	Author(s)
1	2	3	4
		API v.3.2.2. Additional parameter `document_id` have been added to the delivery acceptance method from the Russian Federation `/ships/addImport`, and the mandatory nature of certain attributes has been changed. Section C.5 in Appendix C has been revised.	Matokha
3.9	08.04.2022	Amended and complemented revision of the document describing API v.3.2.2. In the code shipment method `/ships/add` and dairy products shipment recording method `/ships/gtin`, the used type and format for transferring the value of additional information about the price (cost) of the product (`product_cost`) and the amount of VAT (`product_tax`) have been specified.	Valery Matokha
3.10	18.04.2022	Revision of the document describing API v.3.3.0. Goods made from customer's raw materials abelling functionality has been added: - report on the marking of goods from customer's raw materials method `/v2/reports/addMarkTrans` has been added; - new status of abelling code `55` ("Transit goods") has been added; - new type of report `10` ("Labeling of goods made from customer's raw materials") have been added. New section C.6 in Appendix C has been added.	Valery Matokha
3.11	26.05.2022	Revision of the document describing API v.3.4.0. Placement of an order in the printing house for the manufacture of packaging with applied identification means functionality has been added: - creating a request for the manufacture of packaging in a printing house method `/requests/addTypographyOrder` has been added; - receiving a list of requests for the manufacture of packaging in the printing house method `/requests/listTypographyOrders` has been added. In Table D.1 of Appendix D: - for the product group "Dairy products" in the labeling modes "Production", "Import from other countries" and "Import from the EAEU Member States", the list of labeling parameters is	Valery Matokha

Revision	Date	Change description	Author(s)
1	2	3	4
		supplemented with the optional positions “VSD number” and “UI VSD”; - for all product groups in the labeling mode “Import from the EAEU Member States”, the control of the obligatory filling of the labeling parameter “Marking method (print protection)” is set and the use of the labeling parameter “Goods name” is expanded.	
3.12	08.08.2022	Revision of the document describing API v.3.4.1. An additional `name` parameter has been added to the `/contracts/agentsList` method for obtaining a list of counterparties to search by name or part of it. In Table D.1 of Appendix D for all commodity groups in the “Import from other countries” abelling mode: - for the mandatory abelling parameter “Goods declaration registration number”, a format-logical control is established for compliance with the structure of the customs declaration number with the established template; - the list of abelling parameters is supplemented with the optional position “Item number in the goods declaration”. Misprints and inaccuracies have been corrected in the text of the document.	Valery Matokha
3.13	29.09.2022	Revision of the document describing API v.3.4.2. A new method for validating the array of marking codes `/v2/labels` has been added.	Valery Matokha
3.14	31.10.2022	Revision of the document describing API v.3.5.0. The information about the report returned by the `/v2/reports` and `/v2/reports/list` methods is expanded with an additional parameter “`group`”. The possibility of editing abelling reports is implemented: - a new method for adjusting the parameter values of a previously submitted abelling report `/v2/reports/edit` has been added; - a new report type `13` (“Changing marking parameters”) has been added; - expanded examples of response options returned by the `/v2/reports` method.	Valery Matokha
3.15	22.11.2022	Revision of the document describing	Valery

Revision	Date	Change description	Author(s)
1	2	3	4
		API v.3.5.1. The information about the report returned by the `/v2/reports` and `/v2/reports/list` methods is expanded with an additional parameter `document_id`. An additional parameter `document_id` has been added to the `/v2/reports/list` method for obtaining a list of reports to search by the report ID or by the ID of a delivery document from the Russian Federation with the `type = ship`. Expanded examples of response options returned by the `/v2/reports` method.	Matokha
3.16	30.11.2022	Revision of the document describing API v.3.6.0. A new method for getting information about the delivery of dairy products `/ships/getGtin` has been added. The description of the `Ship` object has been expanded in terms of additional fields used in information about the supply of dairy products. The delivery acceptance method from the Russian Federation `/ships/import` and the `Import` object have been modified in terms of the possibility of obtaining additional information about accepted and rejected identification codes.	Valery Matokha
3.17	24.01.2023	Amended and complemented revision of the document describing API v.3.6.0. In the dairy products shipment recording method `/ships/gtin` for the type of operation `operation = export`, the restriction on the quantity of the supplied goods (`count`) has been specified.	Valery Matokha
3.18	20.03.2023	Revision of the document describing API v.3.7.0. In the code shipment method `/ships/add` and dairy products shipment recording method `/ships/gtin`, the requirements for filling in additional information about the unit price of the goods `product_cost` and the amount of VAT `product_tax` have been clarified. The delivery acceptance method from the Russian Federation `/ships/addImport` has been modified: - attribute `document_id` is set to required; - the method of transmitting information with the	Valery Matokha

Revision	Date	Change description	Author(s)
1	2	3	4
		delivery type `type = manual` ("manual entry") is disabled; - trust acceptance is carried out only on the basis of incoming documents with type `type = ship` (cross-border shipment) or `type = export` (register of deliveries according to a temporary scheme) with mandatory indication in the `document_id` parameter of the corresponding UID of the delivery document. Paragraph C.5.2.1 in Appendix C is revised.	
3.19	24.04.2023	Revision of the document describing API v.3.8.0. A new method for checking and verifying the authenticity of marking codes `/v3/checkLabels` has been added.	Valery Matokha
3.20	24.05.2023	Revision of the document describing API v.3.8.1. The features of working with the authorization token returned by the `/auth` method have been clarified. Change password method /users/changePassword defines new password requirements.	Valery Matokha

CONTENTS

ABSTRACT

This document contains the description of the order of cross-system coupling of information systems of participants of merchandise turnover subject to marking with identification means or unified control marks by the hardware- and software system SIS "Electronic Mark" via API protocol.

1 DESIGNATIONS AND ABBREVIATIONS

The following designations and abbreviations are used in this document:

API (Application Programming Interface) is the data exchange protocol, description of the means (a set of classes, procedures, functions, structures or constants) by which a participant's information system can exchange data with the SIS "Electronic Mark";

GTIN (Global Trade Item Number) is the international code of goods marking, identification and accounting;

HTTP (HyperText Transfer Protocol) is the application layer protocol for transferring data between the client and the server; TCP-port 80 is used by default;

HTTPS (HyperText Transfer Protocol Secure) is the extension of the HTTP protocol supporting encryption for enhanced security, data in the HTTPS protocol are transmitted on top of the TLS cryptographic protocols; TCP-port 443 is used by default;

REST (Representational State Transfer) is the architectural style of interaction between components of a distributed application in the network based on the "client-server" model;

TLS (Transport Layer Security) is the cryptographic protocols that enable secure transmission of data between sites on the Internet;

SIS "Electronic Mark" is the state information system of goods marking with the unified control signs or identification means; national component of the goods marking system of the EAEU member state; software and hardware system that ensures the reception, processing, protection, storage and provision of received and processed data on marking goods with identification means, information on operations performed with marked goods;

PC (Protection Character) a document with a certain degree of protection, designed to protect against counterfeiting a physical data carrier of an identification tool that does not contain elements (means) of protection against counterfeiting;

PPDC (Protected Physical Data Carrier), a form of a document with a certain degree of protection, intended for applying, storing and transmitting an identification tool;

Characters – forms of documents and documents with a certain degree of protection, having elements (means) of protection against forgery: unified control signs, protected physical data carriers, protection characters;

EAEU (Eurasian Economic Union) an international organization for regional economic integration with international legal personality and established by the Treaty on the Eurasian Economic Union, the member states of which are the

Republic of Armenia, the Republic of Belarus, the Republic of Kazakhstan, the Kyrgyz Republic and the Russian Federation;

MC (Marking Code) is the representing a unique set of characters used to form identification means generated by the marking system operator;

UCM (Inified Control (identification) Mark) a document with a certain protection class rate used for merchandise marking.

2 GENERAL PROVISIONS

2.1 Interface of cross-system coupling

The web service is designed to enable the interaction of accounting information systems of the turnover participants with the SIS "Electronic Mark" via API protocol, which is implemented within the framework of the client-server REST architecture over a secure HTTPS connection. Privacy of the transmitted data is ensured by TLS security protocol, version 1.2 or higher. Authentication of the turnover participant (user) within the current HTTP session is carried out via a unique API key (authorisation token).

A generalised chart of interaction and data exchange using the "System-to-System" model via the API protocol is shown in Figure 1.

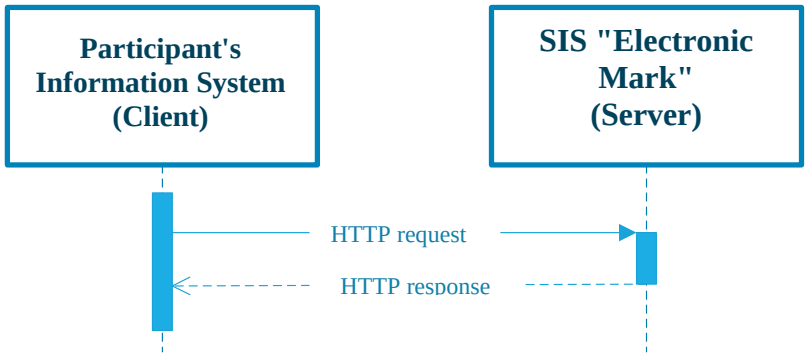


Figure 1 - Chart of cross-system coupling via API protocol

The initiator of HTTP messaging is always the information system or service of the turnover participant.

The information system of the turnover participant acts as a client and sends HTTP queries to the SIS "Electronic Mark", which acts as a server. HTTP queries are transferred between the client and the server. In response to the HTTP request the server sends HTTP response to the client.

By default, the body of the HTTP request and HTTP response is in the format `JSON`.

Data transfer and processing are carried out in the encoding `UTF-8`.

2.2 HTTP request format

The Web Service SIS "Electronic Mark" can be accessed via the following URLs via the API protocol:

Industrial area, productive loop:

`https://api.datamark.by`

Test area, functional testing loop:

`https://v2-sandbox-api.datamark.by`

HTTP queries are sent to the Web Service using the `GET`, `POST` and `PUT` methods (Request Types) over a secure HTTPS connection.

In general, the HTTP request header (`Headers`) shall contain the parameters specified in Table 2.1, unless otherwise specified in the specific API method description.

Table 2.1 - Default HTTP request Header parameters

Key	Meaning
`Content-Type`	`application/json; charset=utf-8`
`Content-Length`	Length of the request body
`Host`	Web service URL (industrial or test area)
`Token`	Authentication token, returned by API method `/auth`

2.3 HTTP response format

The web service returns HTTP response after receiving and processing the HTTP request:

with the response code HTTP 200 — if the HTTP request execution is successful — the HTTP response body is formed in accordance with the structure presented in the description of the corresponding API method;

with the response codes HTTP 4XX — in case of HTTP request execution with an error — HTTP response body generally has the settings listed in Table 2.2.

Table 2.2 - HTTP response settings, Code 4XX

Key	Resp.	Data type	Description
`error`	Yes	Integer	Error code
`message`	Yes	String	User message and/or error description
`details`	No	Array	Detailed error description

Example of a response body with the description of errors in request execution

```
{
    "error": 400,
    "message": "Input parameters validation error",
    "details": [
        "Field ... must be ...",
        "..."
    ]
}
```

Error codes common to all API methods listed in Section 5 of this document.

2.4 Procedure of HTTP queries execution

In the current session, HTTP queries are sent in a sequential order. Each HTTP request activates the appropriate API method, which has the characteristics described in Section 3 of this document:

Request Type (activation method);

Request Line (method name);

set of transferred and returned parameters;

request and response body structure.

All API methods implemented in the Web service are logically combined in the following groups in accordance with their intended use:

user identification methods (`/users`);

methods of working with the goods catalog (`/catalogs`, `/items`);

methods of working with orders (`/orders`);

methods of working with marking codes (`/labels`);

methods of working with reports (`/reports`);

methods of working with counterparties (`/contracts`);

methods of working with shipments (`/ships`);

methods of working with requests (`/requests`);

supplementary methods (`/downloads`, `/directories`).

API methods of new versions are prefixed with `/vN` at the beginning of the Request Line, where `N` is the version number.

The sequence of API methods activation depends on the application environment used (commodity group, type of identification means, marking methods) and the business processes adopted in it. General flow-charts of typical scenarios for the use of API methods when interacting with the SIS "Electronic Mark" is shown:

marking merchandise with means of identification - in Appendix A;

marking merchandise with UCM - in Appendix B.

The specifics of the individual operations of the marking business process for the different commodity groups are given in Appendix B.

2.5 Data types and formats

The following designations of the source data types are adopted In this document:

`String` - string;

`Integer` – integer;

`Float` - a floating-point (comma) number;

`Date` – date and time;

`Boolean` - logical;

`Object` - object;

`Array` - array.

Note: In HTTP requests, the source data types are converted to the relevant data types according to the JSON specification. The following string formats can be used to represent the source data of the `Date` type in JSON structures:

`YYYY-MM-DD hh:mm:ss±hh`;
`YYYY-MM-DD hh:mm:ss`;
`YYYY-MM-DD`.

2.6 Screening

According to JSON specification, invalid characters necessarily shall be screened.

The labelling code and the aggregation code may include one or more unprintable command deliminating character(s) `<GS>` (Group Separator) – code 29 in the table of ASCII characters. The value of unprintable delimiting character `<GS>` in JSON structures shall be represented as screened `\u001d` in Unicode (hereinafter the character is referred to as).

3 API METHODS

3.1 User identification methods

3.1.1 "Authentication" method

Request Type: `POST`
Request Line: `/auth`
Parameters:

Table 3.1.1.1 – HTTP request header parameters

Key	Value
Option 1	
`Content-Type`	`multipart/form-data; boundary=...`
Option 2	
`Content-Type`	`application/x-www-form-urlencoded`

Table 3.1.1.2 - HTTP request parameters

Key	Resp.	Data type	Description
`username`	Yes	String	User e-mail
`password`	Yes	String	User password
`is_rules_agree`	No	Boolean	Indicator of turnover participant's familiarisation and acceptance of the terms of the standard form contract published on the oficial website of the Operator for the implementation of identification means and services for the provision and accounting of marking codes in the SIS "Electronic Mark" (the fact of accession to the standard form contract): `True` - yes; `False` – no. ***Note:*** Specified once at first authentication or if no authentication token is issued in the response

Table 3.1.1.3 - HTTP response parameters, Code 200

Key	Data type	Description
token	String	Authentication token. ***Notes:*** 1. Returns if the turnover participant has read, agreed with and accepted the terms of the standard form contract (`is_rules_agree = True`). 2. The token is valid for 1 hour from the moment of successful authorization. 3. The expiration date and time is specified in the response in the `expires_in` attribute of the `User` object. 4. When the token expires, it is necessary to re-authorize and get a new token.
token_refresh	String	Reserve authorization token. ***Note:*** Not used when working with API methods
User	Object	Authorised user data. Class `User` object (see cl. 4.1.1)

Example of the response body


```
{
    "token":
"eyA0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOjE2ODQ5MTI4OTcsImp0aSI6ImhSR2pJS
1JcL0xEZlVZSjRHTjc1OXFTSUNnVjMyYTk0Y2ZkZGNQUzBiSDdBPSIsImlzcyI6IlwvYXBpIiwibmJm
IjoxNjg0OTEyODk4LCJleHAiOjE2ODQ5MTY0OTcsImRhdGEiOnsiaWQiOjUwNywibmFtZSI6Im1hdG9
oYXN2aXRva0BiYnYuYnkifX0.GQz_nSTUCQJpHP313LQmBLiEPR8Os05Cm-nIeJjhzjK",
    "token_refresh":
"eyA0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOjE2ODQ5MTI4OTcsImp0aSI6IjJOZHp5a
WIwUnFSV0JRTmpNV0tSbDRBSTM0NVZMS2F6RkJEa2x1Rk9Mblk9IiwiaXNzIjoiXC9hcGkiLCJuYmYi
OjE2ODQ5MTI4OTgsImV4cCI6MTY4NDkxNjQ5N30.vPcwOm-6QLZkrBjHaW3AXNFmImGjpJVu-
uIu_PAZsEd",
    "user": {
        "id": 507,
        "name": "Валерий",
        "lastname": "Матоха",
        "email": "matohasvitok@bbv.by",
        "status": 1,
        "expires_in": "2023-05-24 11:21:37",
        "role": {
            "description": "Участник оборота товаров",
            "code": "exporter"
        "info_id": 129,
        "need_change_password": false,
        "info": {
            "name": "СООО \"Свиток\"",
            "address": "г. Минск,ул. Тимирязева, д.2, ком. 16",
            "unp": 190704463,
            "gln": "4811612900000",
            "country": "112",
            "email": "info@bbv.by",
            "is_rules_agree": true
        }
    }
}
```

3.1.2 "Exit" method

Request Type: `POST`
Request Line: `/logout`
Parameters: No input and output parameters

3.1.3 "Password reset" method

Request Type: `POST`
Request Line: `/users/changePassword`
Parameters:

Table 3.1.3.1 - HTTP request parameters

Key	Resp.	Data type	Description
`oldpassword`	Yes	String	Old password
`password`	Yes	String	New password. ***Note:*** The password must meet the following conditions at the same time: 1. Length is not less than 8 characters; 2. Digits, capital and small letters of the Latin alphabet, special symbols are used; 3. Contains at least 1 capital letter, 1 digit and 1 special character.
Example of the request body			

<pre>{
"oldpassword": "********",
"password": "********"
}</pre>			

Table 3.1.3.2 - HTTP response parameters, Code 200

Key	Data type	Description
`status`	String	Indicator of successful operation execution
Example of the response body		

<pre>{
"status": "OK"
}</pre>		

3.2 Methods of working with the goods catalog

3.2.1 "Goods catalog sections" method

Request Type: `POST`
Request Line: `/catalogs/`
Parameters: No input parameters

Table 3.2.1.1 - HTTP response parameters, Code 200

Key	Data type	Description
`catalogs`	Array of Object	Array with a hierarchical structure (tree) of goods catalog sections (categories). Class `Category` objects (see cl. 4.1.3)
Example of the response body (fragment)		
<pre>{</pre>		

```
    "catalogs": [
        {
            "name": "Обувь",
            "group": {
                "code": "shoes",
                "name": "Обувь",
                "articul_required": true,
                "gtin_required": true,
                "is_manually_add": false,
                "is_gtin_add": true,
                "is_show_gtin": true,
                "is_show_articul": true
            },
            "code": 10002
        },
        ...
        {
            "name": "Легкая промышленность",
            "group": {
                "code": "clothes",
                "name": "Легкая промышленность",
                "articul_required": true,
                "gtin_required": true,
                "is_manually_add": false,
                "is_gtin_add": true,
                "is_show_gtin": true,
                "is_show_articul": true
            },
            "code": 10081,
            "inner": [
                {
                    "name": "Блузки, блузы и блузоны трикотажные",
                    "group": {
                        "code": "clothes",
                        "name": "Легкая промышленность",
                        "articul_required": true,
                        "gtin_required": true,
                        "is_manually_add": false,
                        "is_gtin_add": true,
                        "is_show_gtin": true,
                        "is_show_articul": true
                    },
                    "code": 10083
                },
                ...
            ]
        },
        ...
        {
            "name": "Фотоаппараты и лампы-вспышки",
            "group": {
                "code": "photo",
                "name": "Фотоаппараты и лампы-вспышки",
                "articul_required": true,
                "gtin_required": true,
                "is_manually_add": false,
                "is_gtin_add": true,
```

```
                "is_show_gtin": true,
                "is_show_articul": true
            },
            "code": 10088
        },
        ...
        {
            "name": "Товары, подлежащие маркировке УКЗ",
            "group": {
                "code": "ukz",
                "name": "Товары, подлежащие маркировке УКЗ",
                "articul_required": false,
                "gtin_required": false,
                "is_manually_add": false,
                "is_gtin_add": false,
                "is_show_gtin": false,
                "is_show_articul": false
            },
            "code": 10090
        }
    ]
}
```

3.2.2 "Parameters of goods category" method

Request Type: `POST`
Request Line: `/catalogs/{:code}`
Parameters:

Table 3.2.2.1 - HTTP request parameters

Key	Resp.	Data type	Description
`code`	Yes	Integer	Code (identifier) of the goods catalog section (category) derived from the method `/items/catalogs/` (see cl. 3.2.1; the list of values is given in Table 4.1.3.2)
`page`	No	Integer	Page number of the list of goods
Example of the Request Line			
`/catalogs/10089`			
Example of the request body			
`{ "page": 1 }`			

Table 3.2.2.2 - HTTP response parameters, Code 200

Key	Data type	Description
	Object	Goods catalog section (category) Class `Category` object (see cl. 4.1.3)
Example of the response body (fragment)		
Option 1. Dairy products		

```
{
    "code": 10089,
    "name": "Молочная продукция",
    "group": {
```

```
        "code": "milk",
        "name": "Молочная продукция",
        "articul_required": false,
        "gtin_required": true,
        "is_manually_add": false,
        "is_gtin_add": true
    },
    "catalog_params": [
        {
            "group": "Общая характеристика",
            "params": [
                {
                    "type": "text",
                    "name": "ТН ВЭД",
                    "value": {
                        "list": [
                            "0406",
                            "210500",
                            "2202999100",
                            ...
                        ]
                    },
                    "units": null,
                    "code": 40,
                    "is_required": true,
                    "is_editable": false,
                    "is_show_catalog ": false
                },
                {
                    "type": "text",
                    "name": "Тип продукта",
                    "value": {
                        "list": [
                            "МОЛОКО",
                            "МОЛОКО СУХОЕ",
                            "МОЛОКОСОДЕРЖАЩИЙ ПРОДУКТ",
                            "МОЛОКО КОНЦЕНТРИРОВАННОЕ",
                            "МОЛОКО СГУЩЕННОЕ",
                            "МОЛОКОСОДЕРЖАЩИЙ СГУЩЕННЫЙ ПРОДУКТ",
                            ...
                        ]
                    },
                    "units": null,
                    "code": 43,
                    "is_required": true,
                    "is_editable": false,
                    "is_show_catalog ": false
                },
                {
                    "type": "text",
                    "name": "Fat weight content, %",
                    "value": null,
                    "units": null,
                    "code": 46,
                    "is_required": true,
                    "is_editable": false,
                    "is_show_catalog ": false
```

```
                },
                {
                    "type": "select",
                    "name": "Страна происхождения (производства)",
                    "value": {
                        "list": [
                            {
                                "code": "AU",
                                "value": "АВСТРАЛИЯ (AU)"
                            },
                            {
                                "code": "AT",
                                "value": "АВСТРИЯ (AT)"
                            },
                            ...
                        ]
                    },
                    "units": null,
                    "code": 20,
                    "is_required": true,
                    "is_editable": false,
                    "is_show_catalog ": false
                },
                ...
            ]
        },
        {
            "group": "Производитель",
            "params": [
                {
                    "type": "text",
                    "name": "Полное наименование производителя",
                    "value": null,
                    "units": null,
                    "code": 93,
                    "is_required": false,
                    "is_editable": false,
                    "is_show_catalog ": false
                },
                ...
            ]
        },
        ...
    ],
    "mark_params": [
        {
            "group": "Производственная партия",
            "params": [
                {
                    "type": "date",
                    "name": "Yesta изготовления",
                    "value": null,
                    "units": null,
                    "code": 29,
                    "is_required": true,
                    "is_editable": false,
                    "is_show_catalog ": false
```

```
                },
                {
                    "type": "date",
                    "name": "Yesta истечения срока годности",
                    "value": null,
                    "units": null,
                    "code": 122,
                    "is_required": true,
                    "is_editable": false,
                    "is_show_catalog ": false
                },
                {
                    "type": "text",
                    "name": "Переменный вес (значение, ед.изм.)",
                    "value": null,
                    "units": null,
                    "code": 121,
                    "is_required": false,
                    "is_editable": false,
                    "is_show_catalog": false
                },
                {
                    "type": "select",
                    "name": "Тип производственного заказа",
                    "value": {
                        "list": [
                            {
                                "code": "1",
                                "value": "Собственное производство"
                            },
                            {
                                "code": "2",
                                "value": "Контрактное производство"
                            }
                        ]
                    },
                    "units": null,
                    "code": 123,
                    "is_required": false,
                    "is_editable": false,
                    "is_show_catalog": false
                },
                {
                    "type": "select",
                    "name": "Способ маркировки",
                    "value": {
                        "list": [
                            {
                                "code": "1",
                                "value": "Использование защищенного
материального носителя"
                            },
                            {
                                "code": "2",
                                "value": "Использование знака защиты"
                            },
                            {
```

```
                                "code": "3",
                                "value": "Без защиты"
                            }
                        ]
                    },
                    "units": null,
                    "code": 124,
                    "is_required": false,
                    "is_editable": false,
                    "is_show_catalog": false
                },
                {
                    "type": "text",
                    "name": "Партия",
                    "value": null,
                    "units": null,
                    "code": 109,
                    "is_required": true,
                    "is_editable": false,
                    "is_show_catalog ": false
                }
            ]
        }
    ],
    "items": [
    ...
    ]
}
```


Option 2. Goods marked with UCM


```
{
    "code": 10090,
    "name": "Товары, подлежащие маркировке УКЗ",
    "group": {
        "code": "ukz",
        "name": "Товары, подлежащие маркировке УКЗ",
        "articul_required": false,
        "gtin_required": false,
        "is_manually_add": false,
        "is_gtin_add": false,
        "is_show_gtin": false,
        "is_show_articul": false
    },
    "count_items": 22,
    "catalog_params": [],
    "mark_params": [],
    "items": [
        {
            "name": "Икра; консервы рыбные (ТН ВЭД ЕАЭС из 1604)",
            "image": "https://s3.datamark.by/catalogs_sand/876-1624961963-
e3dd0388a6c7fd03f318330176abdec7.jpg",
            "articul": "",
            "code": "100950000583470",
            "description": "",
            "gtin": null,
            "gtin_check": false,
            "group": "ukz",
            "is_my": false,
```

```
            "params": []
        },
        {
            "name": "Воды (ТН ВЭД ЕАЭС из 2202)",
            "image": "https://s3.datamark.by/catalogs_sand/876-1624961879-
036eb7149397ee30252482b443167c26.jpg",
            "articul": "",
            "code": "100950000583469",
            "description": "except 2202 99 910 0, 2202 99 950 0, 2202 99 990
0",
            "gtin": null,
            "gtin_check": false,
            "group": "ukz",
            "is_my": false,
            "params": []
        },
        ...
        {
            "name": "Соки (ТН ВЭД ЕАЭС из 2009)",
            "image": "https://s3.datamark.by/catalogs_sand/876-1624958768-
07a8397f06efab3a7ed5f90ed8b749f5.jpg",
            "articul": "",
            "code": "100950000583449",
            "description": "",
            "gtin": null,
            "gtin_check": false,
            "group": "ukz",
            "is_my": false,
            "params": []
        }
    ]
}
```


3.2.3 "Goods search" method

Request Type: `POST`
Request Line: `/items/findItems`
Parameters:

Table 3.2.3.1 - HTTP request parameters

Key	Resp.	Data type	Description
`search`	Yes	String	Value for the search (GTIN or Trades code/Model)
`is_import`	No	Boolean	Attribute of search through the imported goods catalogue: `True` – search through the imported goods added by the national operators of the EAEU member states; `False` – search through the goods added by turnover agents (default mode). ***Note:*** the defaut mode (`is_import = False`) is applicable where no attribute value is defined.
Example of the request body			

```
{
    "search": "04814888492071"
}
```

Table 3.2.3.2 - HTTP response parameters, Code 200

Key	Data type	Description
`items`	Array of Object	Array containing descriptions of the found goods. Class `Item` objects (see cl. 4.1.4)
Example of the response body		
Option 1. Default search (`is_import = False` or not defined)		

```
{
    "items": [
        {
            "name": "Шина пневматическая  для легковых автомобилей",
            "image": "https://s3.datamark.by/catalogs/404.jpg",
            "articul": "BEL-295",
            "code": "100860000494400",
            "description": "Шина пневматическая  для легковых автомобилей
Белшина 225/65R17   BEL-295   102H  БЕСКАМЕРНЫЕ   ЛЕТНИЕ ШИНЫ",
            "gtin": "04811644004509",
            "gtin_check": true,
            "group": "tires",
            "debug_message": {
                "gtin": "4811644004509",
                "status_code": "1",
                "status_message": "Товар успешно зарегистрирован"
            },
            "is_my": true,
            "catalog": {
                "code": 10086,
                "name": "Шины и покрышки"
            },
            "params": [
                {
                    "name": "Бренд (торговая марка)",
                    "units": null,
                    "code": 22,
                    "value": "Белшина",
                    "display_value": "Белшина"
                },
                {
                    "name": "Страна происхождения (производства)",
                    "units": null,
                    "code": 20,
                    "value": "BY",
                    "display_value": "БЕЛАРУСЬ (BY)"
                },
                {
                    "name": "Наименование товара на этикетке",
                    "units": null,
                    "code": 14,
                    "value": "Шина пневматическая  для легковых автомобилей
225/65R17 BEL-295 102H бескамерная",
                    "display_value": "Шина пневматическая  для легковых
автомобилей 225/65R17 BEL-295 102H бескамерная"
                },
```

```
                {
                    "name": "Сезонность (виды шин)",
                    "units": null,
                    "code": 51,
                    "value": "ЛЕТНИЕ ШИНЫ",
                    "display_value": "ЛЕТНИЕ ШИНЫ"
                },
                {
                    "name": "Ширина профиля шины (камеры)",
                    "units": null,
                    "code": 52,
                    "value": "228",
                    "display_value": "228"
                },
                {
                    "name": "Единица измерения ширины профиля шины",
                    "units": null,
                    "code": 53,
                    "value": "мм",
                    "display_value": "мм"
                },
                {
                    "name": "Номинальное соотношение высоты профиля шины к его
ширине, %",
                    "units": null,
                    "code": 54,
                    "value": "65",
                    "display_value": "65"
                },
                {
                    "name": "Тип конструкции каркаса шины",
                    "units": null,
                    "code": 55,
                    "value": "РАДИАЛЬНЫЙ I",
                    "display_value": "РАДИАЛЬНЫЙ ®"
                },
                {
                    "name": "Номинальный посадочный диаметр обода, дюйм",
                    "units": null,
                    "code": 56,
                    "value": "17",
                    "display_value": "17"
                },
                {
                    "name": "Индекс нагрузки на шину",
                    "units": null,
                    "code": 57,
                    "value": "102",
                    "display_value": "102"
                },
                {
                    "name": "Категория скорости",
                    "units": null,
                    "code": 58,
                    "value": "H",
                    "display_value": "H"
                },
```

```
                {
                    "name": "Тип шины по способу герметизации",
                    "units": null,
                    "code": 59,
                    "value": "БЕСКАМЕРНЫЕ",
                    "display_value": "БЕСКАМЕРНЫЕ"
                },
                {
                    "name": "Код ТН ВЭД ЕАЭС",
                    "units": null,
                    "code": 50,
                    "value": "4011100009",
                    "display_value": "4011100009"
                },
                {
                    "name": "GLN заявителя",
                    "units": null,
                    "code": 42,
                    "value": "4811644900009",
                    "display_value": "4811644900009"
                },
                {
                    "name": "Полное наименование заявителя",
                    "units": null,
                    "code": 37,
                    "value": "Joint-Leftovers Company \"Belshina\"",
                    "display_value": "Joint-Leftovers Company \"Belshina\""
                },
                {
                    "name": "Краткое наименование заявителя",
                    "units": null,
                    "code": 38,
                    "value": "Belshina JSC",
                    "display_value": "Belshina JSC"
                },
                {
                    "name": "Код страны регистрации субъекта заявителя",
                    "units": null,
                    "code": 40,
                    "value": "BY",
                    "display_value": "BY"
                },
                {
                    "name": "GCP заявителя",
                    "units": null,
                    "code": 41,
                    "value": "481164490",
                    "display_value": "481164490"
                },
                {
                    "name": "Идентификатор налогоплательщика (УНП/ИНН)
заявителя",
                    "units": null,
                    "code": 43,
                    "value": "700016217",
                    "display_value": "700016217"
                },
```

```
                {
                    "name": "Адрес заявителя",
                    "units": null,
                    "code": 44,
                    "value": "213824, Республика Беларусь, Могилевская обл., г.
Бобруйск, ш. Минское, д. 4",
                    "display_value": "213824, Республика Беларусь, Могилевская
обл., г. Бобруйск, ш. Минское, д. 4"
                },
                {
                    "name": "GLN производителя",
                    "units": null,
                    "code": 47,
                    "value": "4811644900009",
                    "display_value": "4811644900009"
                },
                {
                    "name": "Полное наименование производителя",
                    "units": null,
                    "code": 49,
                    "value": "Joint-Leftovers Company \"Belshina\"",
                    "display_value": "Joint-Leftovers Company \"Belshina\""
                },
                {
                    "name": "Краткое наименование производителя",
                    "units": null,
                    "code": 48,
                    "value": "Belshina JSC",
                    "display_value": "Belshina JSC"
                },
                {
                    "name": "Код страны регистрации субъекта производителя",
                    "units": null,
                    "code": 45,
                    "value": "BY",
                    "display_value": "BY"
                },
                {
                    "name": "Идентификатор налогоплательщика (УНП/ИНН)
производителя",
                    "units": null,
                    "code": 46,
                    "value": "700016217",
                    "display_value": "700016217"
                }
            ]
        }
    ]
}
```


Option 2. Search through imported goods (`is_import = True`)


```
{
    "items": [
        {
            "unp": "7722735776",
            "name": "Сабо женские, бел/гол, арт. 69630108/06, размер 39",
            "tnved": "6403",
            "gtin": "04640102859104",
```

```
            "brand": "Флоаре",
            "country": {
                "code": "MD",
                "name": "Молдова"
            },
            "is_import": true,
            "params": [
                {
                    "name": "Размер в штихмассовой системе",
                    "display_value": "39",
                    "value": "39"
                },
                {
                    "name": "Вид обуви",
                    "display_value": "САБО",
                    "value": "САБО"
                },
                {
                    "name": "Материал подкладки",
                    "display_value": "без подкладки",
                    "value": "без подкладки"
                },
                {
                    "name": "Материал верха",
                    "display_value": "натуральная кожа",
                    "value": "натуральная кожа"
                },
                {
                    "name": "Цвет",
                    "display_value": "Бел/гол",
                    "value": "Бел/гол"
                },
                {
                    "name": "Материал низа / подошвы",
                    "display_value": "ПВХ",
                    "value": "ПВХ"
                },
                {
                    "name": "Группа ТНВЭД",
                    "display_value": "6403",
                    "value": "6403"
                },
                {
                    "name": "Модель / артикул производителя",
                    "display_value": "69630108/06",
                    "value": "69630108/06"
                },
                {
                    "name": "Тип упаковки",
                    "display_value": "BOX",
                    "value": "BOX"
                },
                {
                    "name": "Код ТНВЭД",
                    "display_value": "-",
                    "value": "-"
                },
```

```
                {
                    "group": "Заявитель",
                    "name": "GCP заявителя",
                    "value": "464010285",
                    "display_value": "464010285"
                },
                {
                    "group": "Заявитель",
                    "name": "GLN заявителя",
                    "value": "4640102859999",
                    "display_value": "4640102859999"
                },
                {
                    "group": "Заявитель",
                    "name": "ИНН заявителя",
                    "value": "7722735776",
                    "display_value": "7722735776"
                },
                {
                    "group": "Заявитель",
                    "name": "КПП заявителя",
                    "value": "772201001",
                    "display_value": "772201001"
                },
                {
                    "group": "Заявитель",
                    "name": "Наименование заявителя",
                    "value": "Общество с ограниченной ответственностью
\"МОЛДШУЗ\"",
                    "display_value": "Общество с ограниченной ответственностью
\"МОЛДШУЗ\""
                },
                {
                    "group": "Заявитель",
                    "name": "ИНН оператора",
                    "value": "7731376812",
                    "display_value": "7731376812"
                },
                {
                    "group": "Заявитель",
                    "name": "Адрес производственной площадки заявителя",
                    "value": "109052, МОСКВА Г, НИЖЕГОРОДСКАЯ УЛ, 104, СТР 7",
                    "display_value": "109052, МОСКВА Г, НИЖЕГОРОДСКАЯ УЛ, 104,
СТР 7"
                },
                {
                    "group": "Производитель",
                    "name": "GCP – глобальный префикс предприятия
производителя",
                    "value": "464010285",
                    "display_value": "464010285"
                },
                {
                    "group": "Производитель",
                    "name": "GLN – глобальный номер места нахождения
производителя",
                    "value": "4640102859999",
```

```
                    "display_value": "4640102859999"
                },
                {
                    "group": "Производитель",
                    "name": "ИНН (или аналог этого кода) производителя",
                    "value": "7722735776",
                    "display_value": "7722735776"
                },
                {
                    "group": "Производитель",
                    "name": "КПП (или аналог этого кода) производителя",
                    "value": "772201001",
                    "display_value": "772201001"
                },
                {
                    "group": "Производитель",
                    "name": "Наименование производителя",
                    "value": "Общество с ограниченной ответственностью
\"МОЛДШУЗ\"",
                    "display_value": "Общество с ограниченной ответственностью
\"МОЛДШУЗ\""
                },
                {
                    "group": "Производитель",
                    "name": "Адрес производственной площадки производителя",
                    "value": "109052, МОСКВА Г, НИЖЕГОРОДСКАЯ УЛ, 104, СТР 7",
                    "display_value": "109052, МОСКВА Г, НИЖЕГОРОДСКАЯ УЛ, 104,
СТР 7"
                }
            ]
        }
    ]
}
```


3.2.4 "Goods addition via GTIN" method

Request Type: `POST`
Request Line: `/items/addByGtin`
Parameters:

Table 3.2.4.1 - HTTP request parameters

Key	Resp.	Data type	Description
`gtin`	Yes	String	GTIN of the added goods. ***Template:*** [0-9]{14}
Example of the request body			

{
"gtin": "04810268043116"
}			

Table 3.2.4.2 - HTTP response parameters, Code 200

Key	Data type	Description
	Object	Description of the added goods. Class `Item` object (see cl. 4.1.4)
Example of the response body		

```
{
    "name": "Сыр",
    "image": "https://s3.datamark.by/catalogs_sand/404.jpg",
    "articul": null,
    "code": "100890000569060",
    "description": "Сыр "Тильзитер" массовая доля жира в сухом веществе 50 %
квадрат, No name, весовой, 8 кг ПФ в г. Барановичи",
    "gtin": "04810268043116",
    "gtin_check": 0,
    "group": "milk",
    "is_my": true,
    "catalog": {
        "code": 10089,
        "name": "Молочная продукция"
    },
    "params": [
        {
            "name": "Идентификатор налогоплательщика (УНП/ИНН) производителя",
            "units": null,
            "code": 96,
            "value": "200030514",
            "display_value": "200030514"
        },
        {
            "name": "Код страны регистрации производителя",
            "units": null,
            "code": 95,
            "value": "BY",
            "display_value": "BY"
        },
        {
            "name": "Краткое наименование производителя",
            "units": null,
            "code": 87,
            "value": "ОАО \"Савушкин продукт\"",
            "display_value": "ОАО \"Савушкин продукт\""
        },
        {
            "name": "Полное наименование производителя",
            "units": null,
            "code": 93,
            "value": "Открытое акционерное общество \"Савушкин продукт\"",
            "display_value": "Открытое акционерное общество \"Савушкин
продукт\""
        },
        {
            "name": "GLN производителя",
            "units": null,
            "code": 97,
            "value": "4810268900006",
            "display_value": "4810268900006"
        },
        {
            "name": "Адрес заявителя",
            "units": null,
            "code": 92,
            "value": "224028, Республика Беларусь, Брестская обл., г. Брест,
```

```
ул. Янки Купалы, д. 118",
            "display_value": "224028, Республика Беларусь, Брестская обл., г.
Брест, ул. Янки Купалы, д. 118"
        },
        {
            "name": "Идентификатор налогоплательщика (УНП/ИНН) заявителя",
            "units": null,
            "code": 91,
            "value": "200030514",
            "display_value": "200030514"
        },
        {
            "name": "GCP заявителя",
            "units": null,
            "code": 89,
            "value": "481026890",
            "display_value": "481026890"
        },
        {
            "name": "Код страны регистрации заявителя",
            "units": null,
            "code": 88,
            "value": "BY",
            "display_value": "BY"
        },
        {
            "name": "Краткое наименование заявителя",
            "units": null,
            "code": 94,
            "value": "ОАО \"Савушкин продукт\"",
            "display_value": "ОАО \"Савушкин продукт\""
        },
        {
            "name": "Полное наименование заявителя",
            "units": null,
            "code": 86,
            "value": "Открытое акционерное общество \"Савушкин продукт\"",
            "display_value": "Открытое акционерное общество \"Савушкин
продукт\""
        },
        {
            "name": "GLN заявителя",
            "units": null,
            "code": 90,
            "value": "4810268900006",
            "display_value": "4810268900006"
        },
        {
            "name": "ТН ВЭД",
            "units": null,
            "code": 40,
            "value": "0406909909",
            "display_value": "0406909909"
        },
        {
            "name": "Масса нетто или объем молочной продукции",
            "units": null,
```

```
            "code": 47,
            "value": "8",
            "display_value": "8"
        },
        {
            "name": "Единица измерения массы нетто или объема молочной
продукции",
            "units": null,
            "code": 106,
            "value": "Килограмм",
            "display_value": "Килограмм"
        },
        {
            "name": "Состав",
            "units": null,
            "code": 50,
            "value": "молоко коровье нормализованное пастеризованное, закваска
на основе молочнокислых мезофильных и термофильных бактерий, соль (содержит
агент антислеживающий ферроцианид калия), консервант нитрат натрия, краситель
аннато, уплотнитель хлорид кальция, ферментный препарат животного
происхождения: пепсин, химозин. Без ГМО. Упакован под вакуумом.",
            "display_value": "молоко коровье нормализованное пастеризованное,
закваска на основе молочнокислых мезофильных и термофильных бактерий, соль
(содержит агент антислеживающий ферроцианид калия), консервант нитрат натрия,
краситель аннато, уплотнитель хлорид кальция, ферментный препарат животного
происхождения: пепсин, химозин. Без ГМО. Упакован под вакуумом."
        },
        {
            "name": "Материал упаковки",
            "units": null,
            "code": 60,
            "value": "ПОЛИМЕР",
            "display_value": "ПОЛИМЕР"
        },
        {
            "name": "Тип упаковки",
            "units": null,
            "code": 59,
            "value": "ВАКУУМНАЯ УПАКОВКА",
            "display_value": "ВАКУУМНАЯ УПАКОВКА"
        },
        {
            "name": "Признак подконтрольности товара ветеринарному надзору",
            "units": null,
            "code": 51,
            "value": "ДА",
            "display_value": "ДА"
        },
        {
            "name": "Признак лечебного или иного специализированного питания на
молочной основе",
            "units": null,
            "code": 49,
            "value": "НЕТ",
            "display_value": "НЕТ"
        },
        {
```

```
            "name": "Признак продукции детского питания на молочной основе",
            "units": null,
            "code": 48,
            "value": "НЕТ",
            "display_value": "НЕТ"
        },
        {
            "name": "Массовая доля белка, %",
            "units": null,
            "code": 105,
            "value": "22",
            "display_value": "22"
        },
        {
            "name": "Fat weight content, %",
            "units": null,
            "code": 46,
            "value": "50",
            "display_value": "50"
        },
        {
            "name": "Происхождение сырья",
            "units": null,
            "code": 44,
            "value": "КОРОВА",
            "display_value": "КОРОВА"
        },
        {
            "name": "Тип продукта",
            "units": null,
            "code": 43,
            "value": "СЫР ВЕСОВОЙ",
            "display_value": "СЫР ВЕСОВОЙ"
        },
        {
            "name": "Наименование товара на этикетке",
            "units": null,
            "code": 100,
            "value": "сыр "Тильзитер" массовая доля  жира в сухом веществе 50
%",
            "display_value": "сыр "Тильзитер" массовая доля  жира в сухом
веществе 50 %"
        },
        {
            "name": "Страна происхождения (производства)",
            "units": null,
            "code": 20,
            "value": "BY",
            "display_value": "БЕЛАРУСЬ (BY)"
        },
        {
            "name": "Товарный знак",
            "units": null,
            "code": 41,
            "value": "Отсутствует",
            "display_value": "Отсутствует"
        }
```

```
    ]
}
```

3.2.5 "Checking GTIN registration status" method

Request Type: `POST`
Request Line: `/items/checkGtin`
Parameters:

Table 3.2.5.1 - HTTP request parameters

Key	Resp.	Data type	Description
`gtin_list`	Yes	Array of String	List of GTINs of the checked goods
`{:gtin}`	Yes	String	GTIN of goods. ***Template:*** [0-9]{14}. ***Limitation:*** Max. 100 GTIN per request
Example of the request body (fragment)			

```
{
    "gtin_list": [
        "04810347004489",
        "04810347004960",
        ...
        "08430854527934"
    ]
}
```

Table 3.2.5.2 - HTTP response parameters, Code 200

Key	Data type	Description
`{:gtin}`	Array of Object	Array with the check results for each GTIN value transferred. The array element key name corresponds to the GTIN, the value of the element contains the check result or an error description
`gtin_check`	Boolean	Indicator of GTIN registration in the catalog: `True` - goods registered; `False` - goods registration pending
Example of the response body (fragment)		

```
{
    "04810347004489": {
        "gtin_check": true
    },
    "04810347004960": {
        "gtin_check": false
    },
    ...
    "08430854527934": {
        "error": 404,
        "message": "Not found"
    }
}
```

3.2.6 "Change of information on goods" method

Request Type: `POST`
Request Line: `/items/edit`
Parameters:

Table 3.2.6.1 - HTTP request parameters

Key	Resp.	Data type	Description
`code`	Yes	String	Internal code (identifier) of the goods in the catalog ***Note***: Only available for products added by a member (i.e. those having `is_my = True`)
`image`	No	String	Product image in Base64 format. Restrictions: 1. Acceptable image file formats are JPG and PNG. 2. File size is no more than 300 Kb. 3. The string must contain an indication of the `MIME-type` `(data:image/jpeg;base64` or `data:image/png;base64).`
`params`	No	Array of Object	Array of additional parameters describing the goods. Composition of the parameters (additional characteristics) and the conditions for their completing are determined by the goods belonging to the relevant catalog section. Class `Param` objects (see cl. 4.1.5). ***Note***: Available only for parameters with values available for editing (i.e., with feature `is_editable = True`)
`code`	Yes	Integer	Parameter code (identifier)
`value`	Yes	String, Integer, Float, Date, Boolean, Object	New parameter value depends on its type

Params

```
{
    "code": "100020000000306",
    "image": " ...
95x8ySorNoCD/2Q==",
    "params": [
        {
            "code": 17,
            "value": "КОЖА"
        },
        {
            "code": 27,
            "value": "Республика Беларусь"
        }
    ]
}
```

Table 3.2.6.2 - HTTP response parameters, Code 200

Key	Data type	Description
`item`	Object	Description of goods with changes and amendments. Class `Item` object (see cl. 4.1.4)

Note: The method is only used to add or change (update) the image of goods.

3.2.7 "Inventory" method

Request Type: `POST`
Request Line: `/v2/items/inventory`
Parameters:

Table 3.2.7.1 –HTTP request parameters

Key	Resp.	Data type	Description
`group`	Yes	String	Commodity group (the list of values is given in Table 4.2.1.2)
`name`	Yes	String	Inventory sheet number
`date`	Yes	Date	Inventory sheet date. ***Format:*** `YYYY-MM-DD`
`quantity`	Yes	Integer	Total quantity of leftovers according to the inventory sheet
Example of the request body			

<pre>{
"group": "shoes",
"name": "124",
"date": "2021-11-01",
"quantity": 3200
}</pre>			

Table 3.2.7.2 – HTTP response parameters, Code 200

Key	Data type	Description
	Object	Inventory sheet. `Inventory` class object (see 4.1.11)
Example of the response body		

<pre>{
"id": "1259835",
"group": "shoes",
"name": "124",
"date": "2021-11-01",
"quantity": 3200
}</pre>		

Note: The usage of the method is possible only when mandatory labeling is introduced and becomes unavailable after the expiration of the time period for transmitting information on the inventory of the left goods established for the corresponding product group.

3.2.8 "Inventory list" method

Request Type: `POST`

Request Line: `/v2/items/inventoryList`
Parameters:

Table 3.2.8.1 – HTTP request parameters

Key	Resp.	Data type	Description
`date_from`	Yes	Date	Search start date. ***Format:*** `YYYY-MM-DD`
`date_to`	Yes	Date	Search end date. ***Format:*** `YYYY-MM-DD`
`group`	Yes	String	Commodity group (the list of values is given in Table 4.2.1.2)
`page`	Yes	Integer	Inventory list page number

Example of the request body

```
{
    "date_from": "2021-11-01",
    "date_to": "2021-11-01",
    "group": "shoes",
    "page": 1
}
```

Table 3.2.8.2 – HTTP response parameters, Code 200

Key	Data type	Description
`count`	Integer	Total number of inventory sheets among search results with the used search parameters (filters)
`inventories_list`	Array of Object	Array containing the inventory sheet data. `Inventory` class objects (see 4.1.11)

Example of the response body

```
{
    "count": 6,
    "inventories_list": [
        {
            "id": "1259835",
            "group": "shoes",
            "name": "124",
            "date": "2021-11-01",
            "quantity": 3200
        },
        {
            "id": "1259834",
            "group": "shoes",
            "name": "123",
            "date": "2021-11-01",
            "quantity": 1875
        }
    ]
}
```

3.3 Methods of working with orders

3.3.1 "Order of marking codes" method

Request Type: `POST`

Request Line: `/v2/orders/add`
Parameters:

Table 3.3.1.1 - HTTP request parameters

Key	Resp.	Data type	Description
`label_type`	Yes	Integer	Code type. Code ID received from the method `/labels/types` (see cl. 3.4.4; the list of values is given in Table 4.2.2.2) ***Limitation:*** Only codes with type `label_type = 6` are available for ordering
`count`	Yes	Integer	Number of codes ordered. ***Limitation***: Not more than 100,000 codes per 1 order
`comment`	No	String	Comment to the order
Example of the request body			

{
"label_type": 6,
"count": 12,
"comment": "Заказ кодов SSCC в количестве 12 (двенадцати) единиц"
}			

Table 3.3.1.2 - HTTP response parameters, Code 200

Key	Data type	Description
`order`	Object	Information on the order. Class `Order` object (see cl. 4.1.6)
Example of the response body		

{
"order": {
"id": "268865",
"created_at": "2021-05-28 14:10:15",
"count": 12,
"gtin": null,
"completed_at": null,
"status": {
"code": 0,
"message": "Создан"
},
"type": {
"id": 6,
"name": "Код агрегации (SSCC)",
"method": "sscc",
"digits_count": null,
"is_bso": false,
"letters_count": null,
"is_autotake": false
}
}
}		

3.3.2 "Obtain signatures to serial numbers" method

Request Type: `POST`
Request Line: `/v2/orders/signatures`

Parameters:

Table 3.3.2.1 - HTTP request parameters

Key	Resp.	Data type	Description
group	Yes	String	Commodity group (the list of values is given in Table 4.2.1.2)
gtin	Yes	Array	List of goods GTINs and their associated serial numbers. ***Limitation:*** Not more than 100 GTINs per 1 request and not more than 10,000 serial numbers for each GTIN in the request
{:gtin}	Yes	Array of String	Key name corresponds to the GTIN value; the item value contains an array of serial numbers associated with the specified GTIN ***Template:*** [0-9]{14}
Example of the request body (fragment)			

```
{
    "group": "tires",
    "gtin": {
        "04811644006879": [
            "ryy322zarvkby",
            "yhgpvzrsqused",
            "atrgjuunwimue",
            ...
            "bqfh7obuuq2sn"
        ],
        "04811644010685": [
            "bqxsgyfyhzerb",
            "b9fffzhgvrav1",
            "etba8xxhyqkmd"
        ],
        ...
        "04811644018605": [
            "awhoazvuglazd",
            "iubq5jvfezevx"
        ]
    }
}
```

Table 3.3.2.2 - HTTP response parameters, Code 200

Key	Data type	Description
{:gtin}	Array of Object	Array with links to the created orders for each GTIN value transferred. The array element key name corresponds to the GTIN, the element value contains a link to the order
id	Integer	Order number (identifier)
created_at	Date	Date and time of order creation
Example of the response body (fragment)		

```
{
    "04811644006879": {
        "id": 268866,
        "created_at": "2021-05-28 15:25:00+03"
    },
    "04811644010685": {
```

```
        "id": 268867,
        "created_at": "2021-05-28 15:25:00+03"
    },
    ...
    "04811644018605": {
        "id": 268912,
        "created_at": "2021-05-28 15:25:01+03"
    }
}
```


3.3.3 "Ordering information" method

Request Type: `POST`
Request Line: `/v2/orders`
Parameters:

Table 3.3.3.1 - HTTP request parameters

Key	Resp.	Data type	Description
`id`	Yes	String, Integer	Order number. Order ID received from methods `/v2/orders/add`, `/v2/orders/addGroupOrders` or `/v2/orders/signature`
Example of the request body			

{
"id": "250879"
}			

Table 3.3.3.2 - HTTP response parameters, Code 200

Key	Data type	Description
`order`	Object	Information on the order. Class `Order` object (see cl. 4.1.6)
Example of the response body		


```
{
    "order": {
        "id": 250879,
        "created_at": "2020-12-15 10:03:00+03",
        "updated_at": "2020-12-15 10:02:25+03",
        "count": 20,
        "comment": "Order 2",
        "parent_order": null,
        "gtin": "04810268024894",
        "completed_at": "2020-12-15 10:04:03.258072+03",
        "status": {
            "code": 30,
            "message": "Выполнен"
        },
        "type": {
            "id": 12,
            "name": "Код РБ идентификации молочной продукции",
            "method": "nobsomilk",
            "digits_count": null,
            "is_bso": false,
            "letters_count": null,
            "is_autotake": false
```

```
        },
        "file": {
            "filename": "https://api-savushkin.datamark.by/downloads/488-2020-
12-15-10-02-25-vzkhwz.txt",
            "downloads": 0
        },
        "user": {
            "name": null,
            "lastname": null,
            "info": {
                "name": "Открытое акционерное общество \"Савушкин продукт\"",
                "address": "Брестская обл.,г. Брест,ул. Янки Купалы, д.118",
                "unp": 200030514,
                "gln": "4810268900006",
                "country": "112",
                "is_rules_agree": true
            }
        },
        "labels": [
            "0104810268024894212fP8sV0r 93VlTU",
            "0104810268024894212w2RoVr0 93!Y+!",
            "0104810268024894212cVXPq8R 93w=Zo",
            ...
            "01048102680248942125AYXlQy 93Mk_R"
        ]
    }
}
```


3.3.4 "Order list" method

Request Type: `POST`
Request Line: `/orders/list`
Parameters:

Table 3.3.4.1 - HTTP request parameters

Key	Resp.	Data type	Description
`date_from`	No	Date	Interval start date. ***Format:*** `YYYY-MM-DD`
`date_to`	No	Date	Interval end date. ***Format:*** `YYYY-MM-DD`
`date_type`	No	String	Data sampling method: `Created` – by creation date; `Completed` – by completion date
`label_type`	No	Integer	Code type. Code ID received from the method `/labels/types` (see cl. 3.4.4; the list of values is given in Table 4.2.2.2)
`page`	Yes	Integer	Page number of the order list
Example of the request body			

{
"date_from": "2020-11-11",
"date_to": "2020-11-11",
"label_type": 6,
"page": 1
}			

Table 3.3.4.2 - HTTP response parameters, Code 200

Key	Data type	Description
`count`	Integer	Total number of orders in sampling with the current sorts (filters). ***Note:*** The parameter will be excluded in the following API versions. Use the `orders/count` method to obtain the number of orders
`orders_list`	Array of Object	Array containing information on the orders. Class `Order` objects (see cl. 4.1.6)

Example of the response body (fragment)


```
{
    "count": 9,
    "orders_list": [
        {
            "id": 253170,
            "created_at": "2020-11-11 17:21:13+03",
            "updated_at": "2020-11-11 17:21:14+03",
            "count": 1,
            "comment": "Aggregation Codes From Postman (SSCC)",
            "parent_order": null,
            "gtin": null,
            "completed_at": "2020-11-11 17:20:47.640486+03",
            "status": {
                "code": 30,
                "message": "Выполнен"
            },
            "type": {
                "id": 6,
                "name": "Код агрегации (SSCC)",
                "method": "sscc",
                "digits_count": null,
                "is_bso": false,
                "letters_count": null,
                "is_autotake": false
            },
            "file": {
                "filename": "https://api-farma-
sandbox.datamark.by/downloads/507-2020-11-11-17-21-12-rsajvi.txt",
                "downloads": 0
            },
            "user": {
                "id": 507,
                "name": "Валерий",
                "lastname": "Матоха",
                "email": "matohasvitok@bbv.by",
                "status": 1,
                "role": {
                    "description": "Участник оборота товаров",
                    "code": "exporter",
                "info_id": 129,
                "need_change_password": false,
                "info": {
                    "name": "СООО \"Свиток\"",
                    "address": "г. Минск,ул. Тимирязева, д.2, ком. 16",
                    "unp": 190704463,
```

```
                    "gln": "4811612900000",
                    "country": "112",
                    "is_rules_agree": true
                }
            }
        },
        ...
    ]
}
```

3.3.5 "Group order of marking codes" method

Request Type: `POST`
Request Line: `/v2/orders/addGroupOrders`
Parameters:

Table 3.3.5.1 - HTTP request parameters

Key	Resp.	Data type	Description
`group`	Yes	String	Commodity group (the list of values is given in Table 4.2.1.2)
`orders`	Yes	Array of Object	Array of parameters of ordering marking codes and/or aggregation codes
`{:gtin}`	Yes	Array of Object	The array element key name corresponds to the GTIN value, the value of the element contains the ordering parameters for the specified GTIN. ***Template:*** [0-9]{14}. ***Limitation:*** Not more than 100 GTINs per 1 request
`count`	Yes	Integer	Number of codes ordered. ***Limitation:*** Per GTIN in the request, depending on the type of code: 1. For code type `label_type = 12` - no more than 100,000 codes; 2. For other types - no more than 10,000 codes.
`label_type`	Yes	Integer	Code type. Code ID received from the method `/labels/types` (see cl. 3.4.4; the list of values is given in Table 4.2.2.2)
`comment`	No	String	Comment to the order
`typography_id`	No	Integer	The printing house ID obtained by the `/users/typographylist method` (see cl. 3.8.4)
`typography_doc`	No	String	Number and date of the order (contract with the printing house)

Example of the request body (fragment)

```
{
    "group": "milk",
    "orders": {
        "04810263033525": {
```

```
            "count": 23,
            "label_type": 12,
            "comment": "Заказ N 1",
            "typography_id": 2,
            "typography_doc" "Договор № 1 от 02.09.2021"
        },
        "04810268024894": {
            "count": 20,
            "label_type": 12,
            "comment": "Заказ N 2",
            "typography_id": 2,
            "typography_doc" "Договор № 1 от 02.09.2021"
        },
        ...
    }
}
```

Table 3.3.5.2 - HTTP response parameters, Code 200

Key	Data type	Description
{:gtin}	Array of Object	Array of order identifiers for each GTIN value transferred. The array element key name corresponds to the GTIN, the element value contains order identifiers
id	Integer	Order number (identifier)
created_at	Date	Date and time of order creation
Example of the response body (fragment)		

```
{
    "04810263033525": {
        "id": "250878",
        "created_at": "2020-12-15 10:03:00+03:00"
    },
    "04810268024894": {
        "id": "250879",
        "created_at": "2020-12-15 10:03:00+03:00"
    },
    ...
}
```

3.3.6 "List of order statuses" method

Request Type: POST
Request Line: /v2/orders/statusList
Parameters:

Table 3.3.6.1 - HTTP request parameters

Key	Resp.	Data type	Description
label_type	No	Integer	Code type. Code ID received from the method /labels/types (see cl. 3.4.4; the list of values is given in Table 4.2.2.2)
page	Yes	Integer	Page number of the order list
Example of the request body			

```
{
    "label_type": 12,
```

```
    "page": 1
}
```

Table 3.3.6.2 - HTTP response parameters, Code 200

Key	Data type	Description
`orders_list`	Array of Object	Array containing brief information on orders and their current status. Class `Order` objects (see cl. 4.1.6)
Example of the response body (fragment)		

```
{
    "orders_list": [
        {
            "id": 250879,
            "created_at": "2020-12-15 10:03:00+03",
            "count": 20,
            "gtin": "04810268024894",
            "completed_at": "2020-12-15 10:04:03.258072+03",
            "status": {
                "code": 30,
                "message": "Выполнен"
            },
            "type": {
                "id": 12,
                "name": "Код РБ идентификации молочной продукции",
                "method": "nobsomilk",
                "digits_count": null,
                "is_bso": false,
                "letters_count": null,
                "is_autotake": false
            }
        },
        {
            "id": 250878,
            "created_at": "2020-12-15 10:03:00+03",
            "count": 23,
            "gtin": "04810263033525",
            "completed_at": null,
            "status": {
                "code": 35,
                "message": "Отклонен"
            },
            "type": {
                "id": 12,
                "name": "Код РБ идентификации молочной продукции",
                "method": "nobsomilk",
                "digits_count": null,
                "is_bso": false,
                "letters_count": null,
                "is_autotake": false
            }
        },
        ...
    ]
}
```

3.3.7 "Number of completed orders" method

Request Type: `POST`
Request Line: `/orders/count`
Parameters:

Table 3.3.7.1 - HTTP request parameters

Key	Resp.	Data type	Description
`date_from`	No	Date	Interval start date. ***Format:*** `YYYY-MM-DD`
`date_to`	No	Date	Interval end date. ***Format:*** `YYYY-MM-DD`
`date_type`	No	String	Data sampling method: `Created` – by creation date; `Completed` – by completion date
`label_type`	No	Integer	Code type. Code ID received from the method `/labels/types` (see cl. 3.4.4; the list of values is given in Table 4.2.2.2)
`page`	Yes	Integer	Page number of the order list
Example of the request body			

<pre>{
"date_from": "2021-05-28",
"date_to": "2021-05-30",
"label_type": 12,
"page": 1
}</pre>			

Table 3.3.7.2 - HTTP response parameters, Code 200

Key	Data type	Description
`count`	Integer	Total number of completed orders in sampling with the current sorts (filters).
Example of the response body		

<pre>{
"count": 9,
}</pre>		

Note: Updating of data returned by the method `/orders/count` is carried out no sooner than two (2) minutes after the last generation of relevant information.

3.4 Methods of working with codes and characters

3.4.1 "Addition of own aggregation codes" method

Request Type: `POST`
Request Line: `/v2/labels/addAggregate`
Parameters:

Table 3.4.1.1 - HTTP request parameters

Key	Resp.	Data type	Description
`aggregate`	Yes	Array of String	Array (list) of aggregation codes generated by a turnover participant
Example of the request body (fragment)			

```
{
    "aggregate": [
        "0004819018880000003202048113090167243 76 1320111110abcdef",
        "0004819018880000001802048113090167243 76 1320111230987zyx",
        ...
        "0004819018880000005602048113090167243 76 1320111230987d1e"
    ]
}
```

Table 3.4.1.2 - HTTP response parameters, Code 200

Key	Data type	Description
count	Object	Number of aggregation codes (total):
success	Integer	- successfully added;
failed	Integer	- erroneous (rejected)
error	Array of String	Array (list) of rejected aggregation codes with error description

Example of the response body

```
{
    "count": {
        "success": 98,
        "failed": 2
    },
    "error": [
        "010481533023884121wbv2 already exists",
        "8974klsakbdlalk3 already exists",
    ]
}
```

Note: The code type value `label_type = 13` is automatically assigned to the own aggregation codes generated and transferred by the turnover participants.

3.4.2 "Deletion of codes from the aggregation" method

Request Type: `POST`
Request Line: `/labels/aggregateDelete`
Parameters:

Table 3.4.2.1 - HTTP request parameters

Key	Resp.	Data type	Description
aggregate	Yes	String	Aggregation code (SSCC or own)
labels	Yes	Array of String	Array of low-level marking codes or aggregation codes to be excluded from aggregation. ***Limitation:*** The codes must be of the same or similar types; low-level marking codes and aggregation codes must not be used simultaneously on the same grouping level

Example of the request body

```
{
    "aggregate": "0004819018880000001802048113090167243 76 1320111230987zyx",
    "labels": [
        "0104811309016724212R7V7577 93iv4=",
        "010481130901672421 28v8LZVn 936rXO"
    ]
}
```

Table 3.4.2.2 - HTTP response parameters, Code 200

Key	Data type	Description
aggregate	String	Processed aggregation code
count	Object	Number of codes (total):
total	Integer	- remaining in the aggregate;
deleted	Integer	- excluded from the aggregate;
failed	Integer	- erroneous
details	Array of String	Array of erroneous codes
Example of the response body		

{
"aggregate":"00048190188800000018020481130901672437 6 1320111230987zyx",
"count": {
"deleted": 2,
"total": 4,
"failed": 0
}
}		

3.4.3 "Information on code and character" method

Request Type: POST
Request Line: /labels/
Parameters:

Table 3.4.3.1 - HTTP request parameters

Key	Resp.	Data type	Description
label	Yes	String	Check code (UUID, serial number, marking code, aggregation code, UCM form series and number, etc.)
Example of the request body			
Option 1. Serial number			

{
"label": "2mXyQa22"
}			

Option 2. Marking code			

{
"label": "0104811309016724 2128v8LZVn 936rXO"
}			

Option 3. Aggregation code (own)			

{
"label": "01048102680248941120051910004837001221 00009"
}			

Option 4. Series and number of the UCM form			

{
"label": "CAA018512030"
}			

Table 3.4.3.2 - HTTP response parameters, Code 200

Key	Data type	Description
label	Object	Information on the marking code. Class Label object (see cl. 4.1.7)
user	Object	Information on the counterparty (current code

		owner). Class `Agent` object (see cl. 4.1.2)
`item`	Object	Description of goods marked with the specified code. Class `Item` object (see cl. 4.1.4)
`story`	Array of Object	Array of events (actions history). Class `Event` object (see cl. 4.1.10)

Example of the response body

Option 1. Serial number


```
{
    "label": {
        "uuid": "2mXyQa22",
        "snomer": "2mXyQa22",
        "is_my": true,
        "type": {
            "id": 14,
            "name": "Серийный номер для молочной продукции",
            "method": "random8",
            "digits_count": 8,
            "is_bso": false,
            "letters_count": null,
            "is_autotake": false
        },
        "status": {
            "code": 15,
            "message": "Выдан код"
        }
    },
    "user": {
        "name": "СООО \"Свиток\"",
        "address": "г. Минск,ул. Тимирязева, д.2, ком. 16",
        "unp": 190704463,
        "gln": "4811612900000",
        "country": "112",
        "is_rules_agree": true
    },
    "story": [
        {
            "date": "2020-11-19 15:10:37+03:00",
            "status": {
                "code": 15,
                "message": "Выдан код"
            },
            "action": {
                "id": 253593,
                "created_at": "2020-11-19 15:11:20+03",
                "updated_at": "2020-11-19 15:11:20+03",
                "count": 1,
                "comment": "Serial Numbers (14 milk) From Postman",
                "parent_order": null,
                "gtin": null,
                "completed_at": "2020-11-19 15:10:38.183691+03",
                "user": {
                    "id": 507,
                    "name": "Валерий",
                    "lastname": "Матоха",
                    "email": "matohasvitok@bbv.by",
```

```
                    "status": 1,
                    "role": {
                        "description": "Участник оборота товаров",
                        "code": "exporter"
                    "info_id": 129,
                    "need_change_password": false,
                    "info": {
                        "name": "СООО \"Свиток\"",
                        "address": "г. Минск,ул. Тимирязева, д.2, ком. 16",
                        "unp": 190704463,
                        "gln": "4811612900000",
                        "country": "112",
                        "is_rules_agree": true
                    }
                },
                "type": {
                    "id": 14,
                    "name": "Серийный номер для молочной продукции",
                    "method": "random8",
                    "digits_count": 8,
                    "is_bso": false,
                    "letters_count": null,
                    "is_autotake": false
                }
            }
        }
    ]
}
```


Option 2. Marking code


```
{
    "label": {
        "uuid":"0104811309016724212 8v8LZVn 936rXO",
        "snomer": "28v8LZVn",
        "is_shipped": true,
        "type": {
            "id": 12,
            "name": "Код РБ идентификации молочной продукции",
            "method": "nobsomilk",
            "digits_count": null,
            "is_bso": false,
            "letters_count": null,
            "is_autotake": false
        },
        "status": {
            "code": 50,
            "message": "Промаркирован"
        }
    },
    "user": {
        "name": "СООО \"Свиток\"",
        "address": "г. Минск,ул. Тимирязева, д.2, ком. 16",
        "unp": 190704463,
        "gln": "4811612900000",
        "country": "112",
        "is_rules_agree": true
    },
    "item": {
```

```
        "name": "Молоко питьевое",
        "image": "https://s3.datamark.by/catalogs_sand/404.jpg",
        "articul": null,
        "code": "100890000567850",
        "description": "Молоко питьевое I,M HAPPY ультрапастеризованное
массовой долей жира 3,3 %, 900 мл, Пюр-пак с крышкой",
        "gtin": "04811309016724",
        "gtin_check": true,
        "group": {
            "code": "milk",
            "name": "Молочная продукция"
        },
        "is_my": true,
        "catalog": {
            "name": "Молочная продукция",
            "group": "milk",
            "code": 10089
        },
        "params": [
            {
                "name": "Идентификатор налогоплательщика (УНП/ИНН)
производителя",
                "units": null,
                "code": 96,
                "value": "400068171",
                "display_value": "400068171"
            },
            {
                "name": "Код страны регистрации производителя",
                "units": null,
                "code": 95,
                "value": "BY",
                "display_value": "BY"
            },
            {
                "name": "Краткое наименование производителя",
                "units": null,
                "code": 87,
                "value": "ОАО \"Милкавита\"",
                "display_value": "ОАО \"Милкавита\""
            },
            {
                "name": "Полное наименование производителя",
                "units": null,
                "code": 93,
                "value": "Открытое акционерное общество \"Милкавита\"",
                "display_value": "Открытое акционерное общество \"Милкавита\""
            },
            {
                "name": "GLN производителя",
                "units": null,
                "code": 97,
                "value": "4811309900009",
                "display_value": "4811309900009"
            },
            {
                "name": "Адрес заявителя",
```

```
                "units": null,
                "code": 92,
                "value": "246029, Республика Беларусь, Гомельская обл., г.
Гомель, ул. Братьев Лизюковых, д. 1",
                "display_value": "246029, Республика Беларусь, Гомельская обл.,
г. Гомель, ул. Братьев Лизюковых, д. 1"
            },
            {
                "name": "Идентификатор налогоплательщика (УНП/ИНН) заявителя",
                "units": null,
                "code": 91,
                "value": "400068171",
                "display_value": "400068171"
            },
            {
                "name": "GCP заявителя",
                "units": null,
                "code": 89,
                "value": "481130990",
                "display_value": "481130990"
            },
            {
                "name": "Код страны регистрации заявителя",
                "units": null,
                "code": 88,
                "value": "BY",
                "display_value": "BY"
            },
            {
                "name": "Краткое наименование заявителя",
                "units": null,
                "code": 94,
                "value": "ОАО \"Милкавита\"",
                "display_value": "ОАО \"Милкавита\""
            },
            {
                "name": "Полное наименование заявителя",
                "units": null,
                "code": 86,
                "value": "Открытое акционерное общество \"Милкавита\"",
                "display_value": "Открытое акционерное общество \"Милкавита\""
            },
            {
                "name": "GLN заявителя",
                "units": null,
                "code": 90,
                "value": "4811309900009",
                "display_value": "4811309900009"
            },
            {
                "name": "ТН ВЭД",
                "units": null,
                "code": 40,
                "value": "0401209109",
                "display_value": "0401209109"
            },
            {
```

```
                "name": "Масса нетто или объем молочной продукции",
                "units": null,
                "code": 47,
                "value": "900",
                "display_value": "900"
            },
            {
                "name": "Единица измерения массы нетто или объема молочной
продукции",
                "units": null,
                "code": 106,
                "value": "Миллилитр",
                "display_value": "Миллилитр"
            },
            {
                "name": "Состав",
                "units": null,
                "code": 50,
                "value": "нормализованное молоко",
                "display_value": "нормализованное молоко"
            },
            {
                "name": "Материал упаковки",
                "units": null,
                "code": 60,
                "value": "БУМАГА – ДРУГОЕ",
                "display_value": "БУМАГА – ДРУГОЕ"
            },
            {
                "name": "Тип упаковки",
                "units": null,
                "code": 59,
                "value": "ПАКЕТ",
                "display_value": "ПАКЕТ"
            },
            {
                "name": "Признак подконтрольности товара ветеринарному
надзору",
                "units": null,
                "code": 51,
                "value": "ДА",
                "display_value": "ДА"
            },
            {
                "name": "Признак лечебного или иного специализированного
питания на молочной основе",
                "units": null,
                "code": 49,
                "value": "НЕТ",
                "display_value": "НЕТ"
            },
            {
                "name": "Признак продукции детского питания на молочной
основе",
                "units": null,
                "code": 48,
                "value": "НЕТ",
```

```
                "display_value": "НЕТ"
            },
            {
                "name": "Массовая доля белка, %",
                "units": null,
                "code": 105,
                "value": "3",
                "display_value": "3"
            },
            {
                "name": "Fat weight content, %",
                "units": null,
                "code": 46,
                "value": "3.3",
                "display_value": "3.3"
            },
            {
                "name": "Происхождение сырья",
                "units": null,
                "code": 44,
                "value": "КОРОВА",
                "display_value": "КОРОВА"
            },
            {
                "name": "Тип продукта",
                "units": null,
                "code": 43,
                "value": "Молоко",
                "display_value": "Молоко"
            },
            {
                "name": "Наименование товара на этикетке",
                "units": null,
                "code": 100,
                "value": "Молоко питьевое ультрапастеризованное массовой долей
жира 3,3 %",
                "display_value": "Молоко питьевое ультрапастеризованное
массовой долей жира 3,3 %"
            },
            {
                "name": "Страна происхождения (производства)",
                "units": null,
                "code": 20,
                "value": "BY",
                "display_value": "БЕЛАРУСЬ (BY)"
            },
            {
                "name": "Товарный знак",
                "units": null,
                "code": 41,
                "value": "I,M HAPPY",
                "display_value": "I,M HAPPY"
            }
        ],
        "mark_params": [
            {
                "name": "Yesta изготовления",
```

```
                "units": null,
                "code": 29,
                "value": "2020-11-22",
                "display_value": "22.11.2020"
            },
            {
                "name": "Партия",
                "units": null,
                "code": 109,
                "value": "19774-П15",
                "display_value": "19774-П15"
            },
            {
                "name": "Переменный вес (значение, ед.изм.)",
                "units": null,
                "code": 121,
                "value": "900 мл",
                "display_value": "900 мл"
            },
            {
                "name": "Yesta истечения срока годности",
                "units": null,
                "code": 122,
                "value": "2021-05-22",
                "display_value": "22.05.2021"
            },
            {
                "name": "Тип производственного заказа",
                "units": null,
                "code": 123,
                "value": "2",
                "display_value": "Контрактное производство"
            },
            {
                "name": "Способ маркировки",
                "units": null,
                "code": 124,
                "value": "1",
                "display_value": "Использование защищенного материального
носителя"
            }
        ]
    },
    "story": [
        {
            "date": "2020-11-18 15:11:17+03:00",
            "status": {
                "code": 15,
                "message": "Выдан код"
            },
            "action": {
                "id": 253489,
                "created_at": "2020-11-18 15:11:58+03",
                "updated_at": "2020-11-18 15:11:59+03",
                "count": 12,
                "comment": "Serial Numbers (14 milk) From Postman",
                "parent_order": null,
```

```
                "gtin": null,
                "completed_at": "2020-11-18 15:11:18.698713+03",
                "user": {
                    "name": "Валерий",
                    "lastname": "Матоха",
                    "info": {
                        "name": "СООО \"Свиток\"",
                        "address": "г. Минск,ул. Тимирязева, д.2, ком. 16",
                        "unp": 190704463,
                        "gln": "4811612900000",
                        "country": "112",
                        "is_rules_agree": true
                    }
                },
                "type": {
                    "id": 14,
                    "name": "Серийный номер для молочной продукции",
                    "method": "random8",
                    "digits_count": 8,
                    "is_bso": false,
                    "letters_count": null,
                    "is_autotake": false
                }
            }
        },
        {
            "date": "2020-11-19 14:01:02+03:00",
            "status": {
                "code": 40,
                "message": "Подписан"
            }
        },
        {
            "date": "2020-11-23 18:50:52+03:00",
            "status": {
                "code": 50,
                "message": "Промаркирован"
            },
            "action": {
                "user": {
                    "name": "Валерий",
                    "lastname": "Матоха",
                    "info": {
                        "name": "СООО \"Свиток\"",
                        "address": "г. Минск,ул. Тимирязева, д.2, ком. 16",
                        "unp": 190704463,
                        "gln": "4811612900000",
                        "country": "112",
                        "is_rules_agree": true
                    }
                }
            }
        },
        {
            "date": "2020-11-24 16:46:50+03:00",
            "status": {
                "code": 400,
```

```
                "message": "Отгружен"
            },
            "action": {
                "name": "ЮЯ03456789",
                "count": 3,
                "comment": "Отгрузка согласно счёт-фактуре № 12456 от
04.11.2020",
                "created_at": "2020-11-24 16:47:34+03",
                "real_count": 15,
                "agg_count": 2,
                "taked_at": "2020-11-24 16:47:02+03",
                "received_count": null,
                "price": {
                    "price": 4567.89,
                    "currency": "rub643"
                },
                "contract": false,
                "operation_date": "2020-11-24",
                "shipping_doc": {
                    "name": "ТН  - товарная накладная",
                    "code": "tnttn"
                },
                "code": 4355,
                "user": {
                    "id": 507,
                    "name": "Валерий",
                    "lastname": "Матоха",
                    "email": "matohasvitok@bbv.by",
                    "status": 1
                },
                "user_from": {
                    "name": "СООО \"Свиток\"",
                    "address": "г. Минск,ул. Тимирязева, д.2, ком. 16",
                    "unp": 190704463,
                    "gln": "4811612900000",
                    "country": "112",
                    "is_rules_agree": true
                },
                "user_to": {
                    "agent": 18413,
                    "name": "ООО \"УНИВЕРСАЛ-ТРЕЙДИНГ\"",
                    "unp": "0274068089",
                    "country": "643",
                    "address": "ул. Милашенкова, д. 4 корп. 7, г. Москва",
                    "gln": "4630019409950"
                },
                "country": {
                    "id": "643",
                    "name": "Россия"
                }
            }
        }
    ]
}
```


Option 3. Aggregation code (own)


```
{
    "label": {
```

```
        "uuid": "0104810268024894112005191000483700122100009",
        "snomer": null,
        "is_shipped": true,
        "is_aggregate": true,
        "agg_count": 4,
        "type": {
            "id": 13,
            "name": "Код агрегации(импорт)",
            "method": "yourself",
            "digits_count": null,
            "is_bso": false,
            "letters_count": null,
            "is_autotake": false
        },
        "status": {
            "code": 30,
            "message": "Валидирован"
        }
    },
    "user": {
        "name": "Открытое акционерное общество \"Савушкин продукт\"",
        "address": "Брестская обл., г. Брест, ул. Янки Купалы, д.118",
        "unp": 200030514,
        "gln": "4810268900006",
        "country": "112",
        "is_rules_agree": true
    },
    "story": [
        {
            "date": "2020-11-17 13:36:40+03:00",
            "status": {
                "code": 20,
                "message": "Изготовлен"
            }
        },
        {
            "date": "2020-11-17 13:36:56+03:00",
            "status": {
                "code": 300,
                "message": "Код агрегирования"
            },
            "total_count": 4,
            "add_count": 4
        },
        {
            "date": "2020-11-18 15:47:02+03:00",
            "status": {
                "code": 400,
                "message": "Отгружен"
            },
            "action": {
                "name": "НЮ7169438",
                "count": 9,
                "comment": "",
                "created_at": "2020-11-18 15:46:21+03",
                "real_count": 109,
                "agg_count": 9,
```

```
                "taked_at": "2020-11-18 15:47:06+03",
                "received_count": null,
                "price": {
                    "price": 508.17,
                    "currency": "byn933"
                },
                "contract": false,
                "operation_date": "2020-11-18",
                "shipping_doc": {
                    "name": "ТТН  - товарно-транспортная накладная",
                    "code": "ttn"
                },
                "code": 4323,
                "user": {
                    "id": 412,
                    "name": "d1",
                    "lastname": null,
                    "email": "dm1@pda.savushkin.by",
                    "status": 1
                },
                "user_from": {
                    "name": "Открытое акционерное общество \"Савушкин
продукт\"",
                    "address": "Брестская обл., г. Брест, ул. Янки Купалы,
д.118",
                    "unp": 200030514,
                    "gln": "4810268900006",
                    "country": "112",
                    "is_rules_agree": true
                },
                "user_to": {
                    "agent": 18399,
                    "name": "Евроторг ООО Ф-Л в г.Бресте",
                    "unp": "201019796",
                    "country": "112",
                    "address": null,
                    "gln": ""
                },
                "country": {
                    "id": "112",
                    "name": "Беларусь"
                }
            }
        }
    ]
}
```


Option 4. Series and number of the UCM form


```
{
    "label": {
        "uuid": "01948104319000002101851203OCAA",
        "snomer": "CAA018512030",
        "info": {
            "created_at": "2021-07-02 00:00:00+03",
            "sorder": "",
            "description": ""
        },
        "is_shipped": true,
```

```
        "type": {
            "id": 17,
            "name": "Унифицированный контрольный знак (18*18)",
            "method": "ukz",
            "digits_count": 9,
            "is_bso": true,
            "letters_count": 3,
            "is_autotake": true
        },
        "status": {
            "code": 30,
            "message": "Валидирован"
        }
    },
    "user": {
        "name": "УП \"Бумажная фабрика\" Гознака г.Борисов",
        "address": "ул. Заводская, 55, г.Борисов, Минской обл., 222516",
        "unp": 600017868,
        "gln": "4810431900000",
        "country": "112",
        "is_rules_agree": true
    },
    "story": [
        {
            "date": "2021-07-02 10:15:50.922458+03",
            "status": {
                "code": 15,
                "message": "Выдан код"
            },
            "action": {
                "id": 275015,
                "created_at": "2021-07-02 10:15:38+03",
                "updated_at": "2021-07-02 10:15:53+03",
                "count": 10000,
                "comment": "",
                "parent_order": null,
                "gtin": null,
                "completed_at": "2021-07-02 10:15:53.496795+03",
                "user": {
                    "name": "Виктор",
                    "lastname": "Соболь",
                    "info": {
                        "name": "УП \"Бумажная фабрика\" Гознака г.Борисов",
                        "address": "ул. Заводская, 55, г.Борисов, Минской обл.,
222516",
                        "unp": 600017868,
                        "gln": "4810431900000",
                        "country": "112",
                        "is_rules_agree": true
                    }
                },
                "type": {
                    "id": 17,
                    "name": "Унифицированный контрольный знак (18*18)",
                    "method": "ukz",
                    "digits_count": 9,
                    "is_bso": true,
```

```
                    "letters_count": 3,
                    "is_autotake": true
                }
            }
        },
        {
            "date": "2021-07-02 10:20:06.172521+03",
            "status": {
                "code": 20,
                "message": "Изготовлен"
            },
            "action": {
                "user": {
                    "name": "Виктор",
                    "lastname": "Соболь",
                    "info": {
                        "name": "УП \"Бумажная фабрика\" Гознака г.Борисов",
                        "address": "ул. Заводская, 55, г.Борисов, Минской обл.,
222516",
                        "unp": 600017868,
                        "gln": "4810431900000",
                        "country": "112",
                        "is_rules_agree": true
                    }
                }
            }
        },
        {
            "date": "2021-07-02 10:33:03.728208+03",
            "status": {
                "code": 400,
                "message": "Отгружен"
            },
            "action": {
                "name": "ФК",
                "count": 50000,
                "comment": "",
                "created_at": "2021-07-02 10:32:12+03",
                "real_count": 50006,
                "agg_count": 1,
                "taked_at": "2021-07-02 10:33:10+03",
                "received_count": null,
                "price": {
                    "price": "",
                    "currency": "byn933"
                },
                "contract": false,
                "operation_date": "2021-06-29",
                "shipping_doc": {
                    "name": "ТТН – товарно-транспортная накладная",
                    "code": "ttn"
                },
                "code": 6916,
                "user": {
                    "name": "Виктор",
                    "lastname": "Соболь",
                    "info": {
```

```
                        "name": "УП \"Бумажная фабрика\" Гознака г.Борисов",
                        "address": "ул. Заводская, 55, г.Борисов, Минской обл.,
222516",
                        "unp": 600017868,
                        "gln": "4810431900000",
                        "country": "112",
                        "is_rules_agree": true
                    }
                },
                "user_from": {
                    "name": "УП \"Бумажная фабрика\" Гознака г.Борисов",
                    "address": "ул. Заводская, 55, г.Борисов, Минской обл.,
222516",
                    "unp": 600017868,
                    "gln": "4810431900000",
                    "country": "112",
                    "is_rules_agree": true
                },
                "user_to": {
                    "name": "Издательство \"Белбланкавыд\" (проверка)",
                    "address": "220037, г. Минск, ул.Ботаническая, 6а",
                    "unp": 100083608,
                    "gln": "4819001820007",
                    "country": "112",
                    "is_rules_agree": true,
                    "agent": 41
                },
                "country": {
                    "id": "112",
                    "name": "Беларусь"
                }
            }
        }
    ]
}
```


3.4.4 "Types of codes and characters" method

Request Type: `GET`
Request Line: `/labels/types`
Parameters: No input parameters

Table 3.4.4.1 - HTTP response parameters, Code 200

Key	Data type	Description
`types_list`	Array of Object	An array with a list of code types. Class `LabelType` object (see cl. 4.2.2)
Example of the response body (fragment)		


```
{
    "types_list": [
        ...
        {
            "id": 6,
            "name": "Код агрегации (SSCC)",
            "method": "sscc",
            "digits_count": null,
```

```
            "is_bso": false,
            "letters_count": null,
            "is_autotake": false
        },
        {
            "id": 7,
            "name": "Код стран ЕАЭС идентификации товара",
            "method": "km",
            "digits_count": null,
            "is_bso": false,
            "letters_count": null,
            "is_autotake": false
        },
        ...
        {
            "id": 12,
            "name": "Код РБ идентификации молочной продукции",
            "method": "nobsomilk",
            "digits_count": null,
            "is_bso": false,
            "letters_count": null,
            "is_autotake": false
        },
        ...
        {
            "id": 20,
            "name": "Код РБ идентификации товара (обувь, шины, легкая
промышленность)",
            "method": "km_rb",
            "digits_count": null,
            "is_bso": false,
            "letters_count": null,
            "is_autotake": false
        }
    ]
}
```


3.4.5 "Cancellation of codes and characters" method

Request Type: `POST`
Request Line: `/labels/destroy`
Parameters:

Table 3.4.5.1 - HTTP request parameters

Key	Resp.	Data type	Description
`shipping_doc`	Yes	String	Type of the document confirming the cancellation. Enumeration value code `shippingDocs` received by the method `/directories/{:code}` (see cl. 3.8.3; the list of values is given in Table 4.3.3)
`nomer_doc`	Yes	String	Number of the document confirming the cancellation
`reason`	Yes	String	Reason for cancellation. Enumeration value code `reasons` received by the method

			`/directories/{:code}` (see cl. 3.8.3; the list of values is given in Table 4.3.4)
`item_name`	No	String	Product name
`period`	No	String	Period (quarter) the forms have been used in
`comment`	No	String	Comment to the cancellation operation
`labels`	Yes	Array of String	Array of marking codes, aggregation codes or series and numbers of UCM forms (characters) subject to cancellation

Example of the request body (fragment)

Option 1. Discarding of marking codes

```
{
    "shipping_doc": "akt",
    "nomer_doc": "LA0232094",
    "reason": "lost",
    "comment": "Утеря кодов маркировки",
    "labels": [
        "0104811309016724212R7V757793iv4=",
        "0104811309016724212v8LZVn936rXO",
        "0104811309016724212FGFq4tJ93m_c9"
        ...
        "0104811309016724212f4DSzvdj936Dyt"
    ]
}
```

Option 2. Discarding of used protection characters

```
{
    "shipping_doc": "akt",
    "nomer_doc": "428",
    "reason": "usemark",
    "item_name": "Обувь мужская",
    "period": "IV квартал 2021 г.",
    "comment": "Использование знаков защиты",
    "labels": [
        "AA012300001",
        "AA012300002",
        "AA012300003",
        ...
        "AA012301000"
    ]
}
```

Table 3.4.5.2 - HTTP response parameters, Code 200

Key	Data type	Description
`message`	String	Informative message
`count`	Integer	Number of cancelled marking, aggregation codes or series and numbers of UCM forms

Example of the response body

```
{
    "message":"СИ успешно списаны",
    "count": 3
}
```

3.4.6 "List of cancellations" method

Request Type: `POST`

Request Line: `/stats/destroyList`
Parameters:

Table 3.4.6.1 – HTTP request parameters

Key	Resp.	Data type	Description
page	Yes	Integer	Page number of the list of cancellations
reason	No	Array of String	Data array containing the list of cancellation reasons
Example of the request body			

```
{
    "page": 1,
    "reason": [
        "damage",
        "lost",
        "usemark"
    ]
}
```

Table 3.4.6.2 - HTTP response parameters, Code 200

Key	Data type	Description
destroy_list	Array of Object	Array with the list of marking codes, aggregation codes or series and numbers of UCM forms cancellations
Example of the response body		

```
{
    "destroy_list": [
        {
            "created_at": "2021-06-17 16:28:41+03",
            "message": {
                "name": "ФН254",
                "reason": "damage",
                "message": "",
                "shipping_doc": "akt"
            },
            "user": {
                "name": "Александр",
                "lastname": "Дещиц",
                "info": {
                    "name": "Индивидуальный предприниматель Дещиц Александр
Евгеньевич",
                    "address": "220116, г Минск, проспект \" Газеты Звязда\" 3-
22",
                    "unp": 100412881,
                    "gln": "4819020530000",
                    "country": "112",
                    "is_rules_agree": true
                }
            },
            "status": {
                "code": null,
                "message": null
            },
            "type": {
                "id": 17,
```

```
                "name": "Унифицированный контрольный знак (18*18)",
                "method": "ukz",
                "digits_count": 9,
                "is_bso": true,
                "letters_count": 3,
                "is_autotake": true
            },
            "label": "01948190085700042101105079lAAA"
        },
        {
            "created_at": "2021-06-29 16:37:27+03",
            "message": {
                "name": "C152-54",
                "reason": "lost",
                "message": "",
                "shipping_doc": "akt"
            },
            "user": {
                "name": "Александр",
                "lastname": "Дещиц",
                "info": {
                    "name": "Индивидуальный предприниматель Дещиц Александр
Евгеньевич",
            "address": "220116, г Минск, проспект \" Газеты Звязда\" 3-22",
                    "unp": 100412881,
                    "gln": "4819020530000",
                    "country": "112",
                    "is_rules_agree": true
                }
            },
            "status": {
                "code": null,
                "message": null
            },
            "type": {
                "id": 17,
                "name": "Унифицированный контрольный знак (18*18)",
                "method": "ukz",
                "digits_count": 9,
                "is_bso": true,
                "letters_count": 3,
                "is_autotake": true
            },
            "label": "01948190085700042101105081lAAA"
        },
        {
            "created_at": "2021-06-29 16:38:29+03",
            "message": {
                "name": "AA25487",
                "reason": "usemark",
                "message": "",
                "shipping_doc": "akt"
            },
            "user": {
                "name": "Александр",
                "lastname": "Дещиц",
                "info": {
```

```
                    "name": "Индивидуальный предприниматель Дещиц Александр
Евгеньевич",
                    "address": "220116, г Минск, проспект \" Газеты Звязда\" 3-
22",
                    "unp": 100412881,
                    "gln": "4819020530000",
                    "country": "112",
                    "is_rules_agree": true
                }
            },
            "status": {
                "code": null,
                "message": null
            },
            "type": {
                "id": 17,
                "name": "Унифицированный контрольный знак (18*18)",
                "method": "ukz",
                "digits_count": 9,
                "is_bso": true,
                "letters_count": 3,
                "is_autotake": true
            },
            "label": "01948190085700042101l050813AAA"
        ]
}
```


3.4.7 "Owner change" method

Request Type: `POST`
Request Line: `/labels/changeOwner`
Parameters:

Table 3.4.7.1 – HTTP request parameters

Key	Resp.	Data type	Description
`group`	Yes	String	Commodity group (the list of values is given in Table 4.2.1.2).
`comment`	No	String	Comment to the owner change transaction
`labels`	Yes	Array of String	Array of labelling codes subject to owner change
Example of the request body (fragment)			


```
{
    "group": "shoes",
    "comment": "",
    "labels": [
        "0104811215019130212tSKmLjdjc6N991TEST92testtest... testtest",
        "0104811215019130212 7USNfqUza15o91TEST92testtest... testtest",
        ...
        "0104811215019130212K2HSsx7xaPPj91TEST92testtest... testtest"
    ]
}
```

Table 3.4.7.2 – HTTP response parameters, Code 200

Key	Data type	Description
`count`	Object	Number of labelling codes (results):
`success`	Integer	- successfully processed;
`failed`	Integer	- invalid (rejected)
`details`	Array	Array of invalid labelling codes
Example of the response body (fragment)		

```
{
    "count": {
        "success": 23,
        "failed": 7
    },
    "details": [
        "Label \"0104811215019130212tSKmLjdjc6N991TEST92test\" is not found",
        "Label \"0104811215019130212 7USNfqUza15o91TEST92test\" is not found",
        ...
    ]
}
```

3.4.8 "Validation of marking codes" method

Request Type: `POST`
Request Line: `/v2/labels`
Parameters:

Table 3.4.8.1 – HTTP request parameters

Key	Resp.	Data type	Description
`labels`	Yes	Array of String	Array of marking codes to be checked. ***Limitation:*** Max. 1 000 marks per request.
Example of the request body (fragment)			

```
{
    "labels": [
        "0104811215019130212tSKmLjdjc6N991TEST92testtest... testtest",
        "0104811215019130212 7USNfqUza15o91TEST92testtest... testtest",
        ...
        "0104811215019130212K2HSsx7xaPPj91TEST92testtest... testtest"
    ]
}
```

Table 3.4.8.2 – HTTP response parameters, Code 200

Key	Data type	Description
`labels`	Array of Objects	Array containing validation results for each marking code
`label`	String	Marking code to be checked
`found`	Boolean	Sign (confirmation) of the presence of a marking code in the information system: `True` – confirmed; `False` – not confirmed
`status`	Object	Current status of the marking code. Class `LabelStatus` object (see cl. 4.2.3). ***Note:*** Returned when `found = true`
`is_my`	Boolean	A sign of code ownership:

		`True` – to the current participant; `False` – to another counterparty. ***Note:*** Returned when `found = true`
`type`	Object	Type of code. Class `LabelType` object (see cl. 4.2.2). ***Note:*** Returned when `found = true`
Example of the response body (fragment)		

```
"labels": [
    {
        "label": "0104811215019130212tSKmLjdjc6N9 91TEST 92testtest... test",
        "found": true,
        "status": {
            "code": 30,
            "message": "Валидирован"
        }
        "is_my": true,
        "type": {
            "id": 20,
            "name": "Код РБ идентификации товара (обувь, шины, легкая
промышленность)"
    },
    {
        "label": "0104811215019130212DvkEvnDcmEdF 91TEST 92testtest... test",
        "found": false
    },
    ...
    {
        "label": "0104811215019130212K2HSsx7xaPPj 91TEST 92testtest... test",
        "found": true,
        "status": {
            "code": 50,
            "message": "Промаркирован"
        }
        "is_my": false,
        "type": {
            "id": 20,
            "name": "Код РБ идентификации товара (обувь, шины, легкая
промышленность)"
    }
]
```

3.4.9 "Information on the structure and composition of the aggregate" method

Request Type: `POST`
Request Line: `/labels/aggregateInfo`
Parameters:

Table 3.4.9.1 - HTTP request parameters

Key	Resp.	Data type	Description
`aggregate`	Yes	String	Aggregation code (SSCC or own)
Example of the request body			

```
{
    "aggregate": "00048190085700001285"
```

```
}
```

Table 3.4.9.2 – HTTP response parameters, Code 200

Key	Data type	Description
	Array of Objects	Composition of the aggregate. ***Note:*** Can include aggregates at lower nesting levels
code	String	Aggregation code or marking code
serial	String	Serial number included in the code or series and number of the form (character)
inner	Array of Objects	Composition of the nested aggregate

Example of the response body (fragment)


```
[
    {
        "code": "02048102230292612136813221214371200422112",
        "serial": null,
        "inner": [
            {
                "code": "0104810223029261212P99Gki6 93z&c:",
                "serial": "DAA001052237"
            },
            {
                "code": "0104810223029261212P99GkiS 93BUtr",
                "serial": "DAA001052238"
            },
            ...
            {
                "code": "0104810223029261212P99GoAC 93P(c*",
                "serial": "DAA001053436"
            }
        ]
    },
    {
        "code": "00048116129001002508",
        "serial": null,
        "inner": [
            {
                "code": "00048116129001002515",
                "serial": null,
                "inner": [
                    {
                        "code": "0104810898000848212P99Wto3 93C6iG",
                        "serial": "DAA001052010"
                    },
                    {
                        "code": "0104810898000848212P99WtMA 93kd,<" ,
                        "serial": "DAA001052011"
                    },
                    {
                        "code": "0104810898000848212P99WtMS 93jvxK",
                        "serial": "DAA001052012"
                    },
                    ...
                ]
            },
            {
```

```
                "code": "00048116129001002522",
                "serial": null,
                "inner": [
                    {
                        "code": "0104810268023880212P99St2K 93hM+W",
                        "serial": "0104810268023880212P99St2K"
                    },
                    {
                        "code": "0104810268023880212P99StFh 93rTGY",
                        "serial": "0104810268023880212P99StFh"
                    },
                    {
                        "code": "0104810268023880212P99StET 93u_4a",
                        "serial": "0104810268023880212P99StET"
                    },
                    ...
                ]
            },
            ...
        ]
    },
    ...
    {
        "code": "f5366ad6-d136-11ed-afa1-0242ac120002",
        "serial": null
    }
]
```

3.4.10 "Checking and verifying the authenticity of marking codes" method

Request Type: `POST`
Request Line: `/v3/checkLabels`
Parameters:

Table 3.4.10.1 – HTTP request parameters

Key	Resp.	Data type	Description
`labels`	Yes	Array of String	Array of marking codes to be checked. ***Note:*** Marking codes from different commodity groups are allowed. ***Limitation:*** Max. 100 marks per request
Example of the request body (fragment)			

```
{
    "labels": [
        "0104810268049019212P9l66jN 930C+%",
        "0104810439006605212PPzXsXH 93txN9",
        "DVd;kbeVDVEgegkv+f;wgpegvd;v",
        "0104810005000860215Rt1Sz 93dGVz",
        "0104810005000860215Gckdg 93grED",
        "010481000500086021 58a49O 93TsHj",
        ...
    ]
}
```

Table 3.4.10.2 – HTTP response parameters, Code 200

Key	Data type	Description
`labels`	Array of Objects	Array containing the results of the check for each marking code
`{:label}`	String	The name of the array element key corresponds to the marking code being checked, the value of the element contains the results of checking the given marking code
`type`	Object	Type of code. Class `LabelType` object (see cl. 4.2.2). ***Note:*** Displayed for codes issued in the Republic of Belarus
`validation`	Boolean	Sign of confirmation of the correct structure of the marking code: `True` – confirmed; `False` – not confirmed. ***Note:*** Optional parameter. Output when `validation = false`
`found`	Boolean	Sign of the presence of a marking code in the information system: `True` – marking code found; `False` – the marking code was not found. ***Note:*** Optional parameter. Output when `found = false`
`in_circulation`	Boolean	Sign of the presence of the marked goods in circulation on the territory of the Republic of Belarus: `True` – in circulation; `False` – not in circulation. ***Note:*** Returned when `validation = true` and `found = true`
`status`	Object	Current status of the marking code. ***Notes:*** 1. Displayed when `validation = true` and `found = true`. 2. For marking codes issued in the SIS "Electronic Mark", an object of the `LabelStatus` class is returned (see cl. 4.2.3). 3. For marking codes issued in the national marking systems of other member states of the EAEU, the following values are displayed: `1` – Status of the Russian Federation: in circulation; `2` – Status of the Russian Federation: not in circulation.
`snomer`	String	Series and number of a document with a certain protection level. ***Note:*** Used for all code types with the attribute `is_bso = True`

item	Object	Brief information about the goods. ***Note:*** Returned when `validation = true` and `found = true`
gtin	String	Product GTIN
tnved	String	EAEU HS Code
name	String	Goods name
group	String	Commodity group (the list of values is given in Table 4.2.1.2)
expirationDate	Date	Expiration date. ***Note:*** Displayed only for dairy products `group = milk`

Example of the response body (fragment)


```
{
    "labels": {
        "0104810268049019212P9l66jN 930C+%": {
            "type": {
                "id": 12,
                "name": "Код РБ идентификации молочной продукции",
                "method": "nobsomilk",
                "digits_count": null,
                "is_bso": false,
                "letters_count": null,
                "is_autotake": false
            },
            "in_circulation": false,
            "status": {
                "code": 30,
                "message": "Валидирован"
            },
            "item": {
                "gtin": "04810268049019",
                "tnved": "0406105002",
                "name": "Творог",
                "group": "milk"
            }
        },
        "0104810439006605212PPzXsXH 93txN9": {
            "type": {
                "id": 18,
                "name": "Стикер, содержащий средство идентификации (18*18)",
                "method": "stiker",
                "digits_count": 9,
                "is_bso": true,
                "letters_count": 3,
                "is_autotake": true
            },
            "in_circulation": true,
            "status": {
                "code": 50,
                "message": "Промаркирован товар"
            },
            "snomer": "SAA154747384",
            "item": {
                "gtin": "04810439006605",
                "tnved": "2105009900",
                "name": "Мороженое",
```

```
                "group": "milk",
                "expirationDate": "2023-09-30"
            }
        },
        "DVd;kbeVDVEgegkv+f;wgpegvd;v": {
            "validation": false
        },
        "0104810005000860215Rt1Sz 93dGVz": {
            "found": false
        },
        "0104810005000860215Gckdg 93grED": {
            "item": {
                "gtin": "04810005000860",
                "tnved": "0404908100",
                "name": "Молочый Продуктъ",
                "group": "milk",
                "expirationDate": "2022-12-16"
            },
            "in_circulation": true,
            "status": {
                "code": "1",
                "message": "Статус РФ: в обороте"
            }
        },
        "010481000500086021 58a49O 93TsHj": {
            "item": {
                "gtin": "04810005000860",
                "tnved": "0404908100",
                "name": "Молочый Продуктъ",
                "group": "milk"
            },
            "in_circulation": false,
            "status": {
                "code": "2",
                "message": "Статус РФ: не в обороте"
            }
        },
        ...
    }
}
```


3.5 Methods of working with the reports

3.5.1 "Report on marking" method

Request Type: `POST`
Request Line: `/v2/reports/addMark`
Parameters:

Table 3.5.1.1 - HTTP request parameters

Key	Resp.	Data type	Description
`group`	Yes	String	Commodity group (the list of values is given in Table 4.2.1.2).
`labels`	Yes	Array of String	Array (list) of marking codes applied on the goods.

			Limitations: The codes listed in the array shall meet the following requirements: 1. Labelling codes shall pertain to the same GTIN; 2. Not more than 25,000 marking codes per 1 request
`params`	No	Array of Object	Array of additional parameters describing the batch of marked goods. Composition of the parameters (additional characteristics) and the conditions for their completing are determined by the marked goods belonging to the relevant catalog section. Class `Param` objects (see cl. 4.1.5 and Appendix D)

Example of the request body

```
{
    "group": "milk",
    "labels": [
        "0104811258000850212apa5tyq 93_dwG",
        "0104811258000850212pxq5tkk 93NQw^"
    ],
    "params": [
        {
            "code": 25,
            "value": "2020-11-22"
        },
        {
            "code": 90,
            "value": "19774-P15"
        },
        {
            "code": 91,
            "value": "900 мл"
        },
        {
            "code": 92,
            "value": "2021-05-22"
        },
        {
            "code": 93,
            "value": "2"
        },
        {
            "code": 94,
            "value": "Масло сливочное"
        },
        {
            "code": 95,
            "value": "1"
        }
    ]
}
```

Table 3.5.1.2 - HTTP response parameters, Code 200

Key	Data type	Description
`report_id`	String	UID, unique identifier of the report on

		marking

Example of the response body

```
{
    "report_id": "22331fa0-740c-4307-80a7-10b24827d9ff"
}
```

3.5.2 "Report on group marking with aggregation" method

Request Type: `POST`
Request Line: `/v2/reports/addGroupMark`
Parameters:

Table 3.5.2.1 - HTTP request parameters

Key	Resp.	Data type	Description
`group`	Yes	String	Commodity group (the list of values is given in Table 4.2.1.2)
`packages`	Yes	Array	Array of packages subject to marking and aggregation
`aggregate`	Yes	String	Aggregation code (SSCC or own) applied on the packageing. ***Limitation:*** Not more than 100 aggregation codes per 1 request
`labels`	Yes	Array of String	Array (list) of marking codes applied on the goods included in the package. Limitations: The codes listed in the array shall meet the following requirements: 1. Labelling codes shall pertain to the same GTIN; 2. Not more than 25,000 marking codes per 1 request
`params`	No	Array of Object	Array of additional parameters describing the batch of marked goods. Composition of the parameters (additional characteristics) and the conditions for their completing are determined by the marked goods belonging to the relevant catalog section. Class `Param` objects (see cl. 4.1.5 and Appendix D).

Example of the request body (fragment)

```
{
    "group": "milk",
    "packages": [
        {
            "aggregate": "0104810268024917112009271007653700122100600",
            "labels": [
                "0104811258000850212apa5tyq93_dwG",
                "0104811258000850212pxq5tkk93NQw^"
        },
        {
            "aggregate": "0104810268024917112009271007653700122100601",
            "labels": [
                "0104811258000850212grxc2dd93_2ds",
                "0104811258000850212osbe43938dbw="
        },
```

```
        ...
    ],
    "params": [
        {
            "code": 25,
            "value": "2020-11-22"
        },
        {
            "code": 90,
            "value": "19774-П15"
        },
        {
            "code": 91,
            "value": "900 мл"
        },
        {
            "code": 92,
            "value": "2021-05-22"
        },
        {
            "code": 93,
            "value": "2"
        },
        {
            "code": 94,
            "value": "Масло сливочное"
        },
        {
            "code": 95,
            "value": "1"
        }
    ]
}
```

Table 3.5.2.2 - HTTP response parameters, Code 200

Key	Data type	Description
packages	Array	Array of packages that have passed marking and aggregation
{:aggregate}	Array	The array element key name corresponds to the aggregation code applied on the package, the element value contains the UID of the report on marking
report_id	String	UID, unique identifier of the report on marking
Example of the response body (fragment)		

```
    "packages": {
        "0104810268024917112009271007653 70012 2100600": {
            "report_id": "58ef4625-c8bf-4cdc-8d06-cf9111a36677"
        },
        "0104810268024917112009271007653 70012 2100601": {
            "report_id": "2909732b-a9ca-4a4b-b3bd-3e18b575c9e1"
        },
        ...
    }
}
```

3.5.3 "Report on aggregation" method

Request Type: `POST`
Request Line: `/v2/reports/addAggregate`
Parameters:

Table 3.5.3.1 - HTTP request parameters

Key	Resp.	Data type	Description
`aggregate`	Yes	String	Aggregation code (SSCC or own)
`labels`	Yes	Array of String	Array of marking codes, low-level aggregation codes subject to aggregation. ***Limitations:*** The codes listed in the array must conform to the following conditions: 1. Marking codes must belong to the same commodity group; 2. The marking code must not be a part of another aggregate; 3. The aggregation code must not be "empty", i.e. it must contain nested codes (i.e. subordinate marking codes and/or low-level aggregation codes); 4. The aggregation code must not be a part of a top-level aggregate; 5. It is forbidden to simultaneously specify the marking code and the aggregation code to which it is "linked"; 6. In total, not more than 100 000 marking codes (including those included in the aggregation) per 1 request
Example of the request body			
Option 1. Entry-level aggregation (marking codes)			

```
{
    "aggregate": "00048190188800000018020481130901672437 6 1320111230987zyx",
    "labels": [
        "0104811309016724212FGFq4tJ93m_c9",
        "0104811309016724212Ddivj1593m2dN",
        "0104811309016724212Seyct2y93mKCh"
    ]
}
```

Option 2. Aggregation of the second and subsequent levels (nested aggregation codes)

```
{
    "aggregate": "00048116129000000529",
    "labels": [
        "000481901888000000180204811309016724376 1320111230987zyx",
        "000481901888000000250204811309016724376 1320111230123abc",
        "000481901888000000320204811309016724376 1320111230klm546"
    ]
}
```

Table 3.5.3.2 - HTTP response parameters, Code 200

Key	Data type	Description
`report_id`	String	UID, unique identifier of the report on aggregation

Example of the response body

{
"report_id": "22331fa0-740c-4307-80a7-10b24827d9ff"
}

3.5.4 "Use of UMC" method

Request Type: `POST`
Request Line: `/v2/reports/addUsed`
Parameters:

Table 3.5.4.1 - HTTP request parameters

Key	Resp.	Data type	Description
`group`	Yes	String	The value `"ukz" is indicated` for the UCM-marked goods
`item`	Yes	String	Goods code from the catalog. The `code` parameter from the `items` list of goods obtained by the method `/catalogs/{:code}`, where `code = 10090` which correspond `ukz` product group (see Table 4.1.4.2)
`labels`	Yes	Array of String	Array (list) of the series and numbers of the used UCM. ***Limitation:*** No more than 25 000 characters per 1 request.
Example of the request body (fragment)			

{
"group": "ukz",
"item": "100900000695408", //Кофе (ТН ВЭД ЕАЭС из 0901)
"labels": [
"AAA011050881",
"AAA011050882",
...
"AAA011051000"
]
}			

Table 3.5.4.2 - HTTP response parameters, Code 200

Key	Data type	Description
`report_id`	String	UID, the unique identifier of the UCM usage report
Example of the response body		

{
"report_id": "4cd57fc9-0fef-4857-988d-621e6ecbbedc"
}		

3.5.5 "Report execution status" method

Request Type: `POST`
Request Line: `/v2/reports`
Parameters:

Table 3.5.5.1 - HTTP request parameters

Key	Resp.	Data type	Description
report_id	Yes	String	Unique identifier of the report. UID received from methods /v2/reports/addMark, /v2/reports/addGroupMark, /v2/reports/addAggregate, /v2/reports/addUsed, /v2/reports/addCancelUsed, /v2/reports/addMarkCirculation or /v2/reports/addMarkLeftovers, as well as /v2/reports/list
Example of the request body			
{ "report_id": "22331fa0-740c-4307-80a7-10b24827d9ff" }			

Table 3.5.5.2 - HTTP response parameters, Code 200

Key	Data type	Description
created_at	Date	Date and time of creation of the report
updated_at	Date	Date and time of the last data update or reporting
type	Integer	Report type ID (the list of values is given in Table 4.2.8.2)
report_id	String	UID, unique identification number of a report
group	String	Commodity group (the list of values is given in Table 4.2.1.2). ***Note:*** Available only for reports generated by methods with the "group" input parameter
status	Object	The current status of the report. Class ReportStatus object (see cl. 4.2.5)
document_id	String	ID of a delivery document from the Russian Federation. ***Note:*** Displayed for reports generated by the /v2/reports/addMarkCirculation method, if the specified parameter has been filled in
result	Object	Presentation of the results of report processing
labels	Object	Results of marking codes processing
aggregate	Object	Results of aggregation codes processing
Example of the response body		
Option 1. Report created, but not processed		
{ "created_at": "2020-12-15 09:55:46+03", "updated_at": null, "type": 1, "report_id": "22331fa0-740c-4307-80a7-10b24827d9ff", "group": "tires", "group": "shoes",		

```
    "status": {
        "code": 0,
        "message": "Создан"
    },
    "result": null
}
```


Option 2. Report is complete and without errors


```
{
    "created_at": "2020-12-23 10:48:56+03",
    "updated_at": "2020-12-23 10:48:57.001371+03",
    "type": 1,
    "report_id": "59b8ea17-81f0-47ea-99c4-60eeffe1f347",
    "status": {
        "code": 30,
        "message": "Выполнен"
    },
    "result": {
        "labels": {
            "success": 5
        }
    }
}
```


Option 3. Report on marking with aggregation is complete and without errors


```
 {
     "created_at": "2020-12-23 11:01:38+03",
     "updated_at": "2020-12-23 11:41:40.124931+03",
     "type": 1,
     "report_id": "f84632c1-e574-4ec0-b00d-6937bc457a56",
    "group": "milk",
     "status": {
         "code": 30,
         "message": "Выполнен"
     },
     "result": {
         "labels": {
             "success": 5
         },
         "aggregate": {
             "success": 5
         }
     }
 }
```


Option 4. Report on marking with aggregation is partially completed (marking was carried out but without aggregation)


```
{
    "created_at": "2020-12-23 10:58:28+03",
    "updated_at": "2020-12-23 10:58:28.738633+03",
    "type": 1,
    "report_id": "00849e06-5000-4c8c-9546-def9c8201de3",
    "group": "milk",
    "status": {
        "code": 30,
        "message": "Выполнен"
    },
    "result": {
        "labels": {
```

```
            "success": 5
        },
        "aggregate": "Aggregate not found"
    }
}
```


Option 5. Report on marking with aggregation is partially completed (aggregation is complete, except for the rejected marking codes)


```
{
    "created_at": "2020-12-23 11:02:21+03",
    "updated_at": "2020-12-23 11:02:22.001934+03",
    "type": 1,
    "report_id": "e841883f-b89b-43ee-bad2-4b91f5afa658",
    "group": "milk",
    "status": {
        "code": 30,
        "message": "Выполнен"
    },
    "result": {
        "labels": {
            "failed": 2,
            "details": [
                "0104810263033525212sFV261B935GVA not found",
                "0104810263033525212D2v4Ddv93dF2E not found"
            ],
            "success": 3
        },
        "aggregate": {
            "success": 4
        }
    }
}
```


Option 6. Report with aggregation is partially completed (part of the marking codes are not found)


```
{
    "created_at": "2022-01-21 17:16:20+03",
    "updated_at": "2022-01-21 19:15:03.418405+03",
    "type": 2,
    "report_id": "661acd42-60fa-464e-8348-d186bf80be9e",
    "status": {
        "code": 30,
        "message": "Выполнен"
    },
    "result": {
        "labels": {
            "failed": 10,
            "details": [
                "0104814130000054212 1vtxTq2qgpL5910005921S1EbB8T3A1gtdPY72J
AK9f3ouksrDcgzPtZojqs2YrJ not found",
                "0104814130000054212 32VxgAgX1cLi910005921dwpgPBzHa4J9DDLRTU
x9RhziGVuaY2ticE3di1TVL4t not found",
                ...
               "0104814130000054212 Y33i7WyHbhe3910005922MGuuEi3cKrAnzXmRJ6n
2vpHXBgXoVY7EHXDxxR8CrrH not found"
            ]
        },
        "aggregate": {
            "success": 90
        }
```

```
    }
}
```


Option 7. Report is not completed due to an error


```
{
    "created_at": "2020-12-23 02:30:52+03",
    "updated_at": "2020-12-23 02:30:52.001098+03",
    "type": 1,
    "report_id": "791f0160-f62e-466a-85c8-26e50713a7ab",
    "group": "milk",
    "status": {
        "code": 40,
        "message": "Error"
    },
    "result": {
        "error": 2001,
        "message": "GTIN not found"
    }
}
```


Option 8. Leftovers labelling report is partially completed (a labelling code is not found)


```
{
    "created_at": "2021-10-10 15:00:20+03",
    "updated_at": "2021-10-10 15:00:21.002721+03",
    "type": 6,
    "report_id": "9f7a3581-9447-425a-a931-49412b3fdb19",
    "group": "leftovers_shoes",
    "status": {
        "code": 30,
        "message": "Выполнен"
    },
    "result": {
        "labels": {
            "failed": 1,
            "details": [
                {
                    "label":"0104812700000039212FcDtbAMjwCfY91TEST92test
test...test test not found",
                    "comment": "Новая обувь, партия 127"
                },
                {
                    "label":"0104810978916892212Y6MayiL1LqLY91TEST92test
test...testtest",
                    "comment": "Обувь женская, для вывоза из РБ"
                },
                {
                    "label":"0104814130000030212srxv8YiZ4bpN91TEST92test
test...testtest",
                    "comment": "Рспродажа в Беларуси"
                },
                ...
            ],
            "success": 24
        },
        "gtin_count": {
            "04810978916892": 1,
            "04814130000030": 1,
                ...
        }
```

```
    }
}
```


Option 9. Report on commissioning is completed and without errors


```
{
    "created_at": "2022-01-18 18:55:56+03",
    "updated_at": "2022-01-18 18:55:57.551992+03",
    "type": 7,
    "report_id": "c8303b42-fd5f-4e6d-945f-011fc6125606",
    "group": "clothes",
    "status": {
        "code": 30,
        "message": "Выполнен"
    },
    "result": {
        "labels": {
            "success": 6
        },
        "gtin_count": {
            "04811595000018": 3,
            "08808563305196": 3
        }
    }
}
```


Option 10. Report on putting into circulation is partially completed (the authenticity and acceptable status of part of the marking codes is not confirmed for various reasons)


```
{
    "created_at": "2022-01-20 12:42:02+03",
    "updated_at": "2022-01-20 12:42:02.865741+03",
    "type": 7,
    "report_id": "7a6bcc5c-a9b3-4071-bfd8-e4d42d946a85",
    "group": "clothes",
    "status": {
        "code": 30,
        "message": "Выполнен"
    },
    "result": {
        "labels": {
            "failed": 5,
            "details": [
                {
                    "label":"0104640043462197215JDmqbLtKlYIN91FFD092dGVzdN3w
8G+M+YxgA5vsz98pvQzh078mehnUQ7ByRkA= not found. Не подтверждено оператором. В
обороте."
                },
                ...
                {
                    "label": "0104640043462197152cpfgcqb5pdx91809792co51XTc
PbDgmUbdylDJMFoR1VbVUGYx/JO09x1iePe3hiFxvgXfI2SHS3ycDJiZCwID1LA7zGR8cGg1NpdtBDA
== not found. Не подтверждено оператором. Неверный формат."
                }
            ],
            "success": 95
        },
        "gtin_count": {
            "04640043462197": 95
        }
    }
```

```
}
```

Option 10. Report on putting into circulation was made on the basis of the delivery document from the Russian Federation and the labeling codes were successfully processed


```
{
    "created_at": "2022-02-09 12:43:10+03",
    "updated_at": "2022-02-09 12:43:10.913544+03",
    "type": 7,
    "report_id": "f9a7b2e2-b3c5-4115-a36f-1bd3985a102c",
    "group": "clothes",
    "document_id": "cf1b944d-e1cd-4d45-8544-6905d9eab29b",
    "status": {
        "code": 30,
        "message": "Выполнен"
    },
    "result": {
        "labels": {
            "success": 2
        },
        "gtin_count": {
            "00194501164733": 2
        }
    }
}
```


Option 12. Report on changing the marking parameters of a previously submitted labelling report


```
{
    "created_at": "2022-10-17 11:17:09+03",
    "updated_at": "2022-10-17 11:17:09.633828+03",
    "type": 13,
    "report_id": "67db7b9d-ed03-4347-a849-753eed01076c",
    "status": {
        "code": 30,
        "message": "Выполнен"
    },
    "result": {
        "params": {
            "success": 5
        }
    }
}
```


3.5.6 "Reports list" method

Request Type: `POST`
Request Line: `/v2/reports/list`
Parameters:

Table 3.5.6.1 - HTTP request parameters

Key	Resp.	Data type	Description
`report_type`	Yes	Integer	Report type. Report type ID received from the method `/v2/reports/types` (see cl. 3.5.7; the list of values is given in Table 4.2.8.2)
`document_id`	No	String	UID, unique identifier of the report or unique

			identifier of a delivery document from the Russian Federation (for reports generated by the `/v2/reports/addMarkCirculation` method, if the specified parameter was filled in)

Example of the request body

```
{
    "report_type": 1
}
```

Table 3.5.6.2 - HTTP response parameters, Code 200

Key	Data type	Description
`report_list`	Array of Object	Array with the list of reports statuses. ***Note:*** No more than 100 of the last generated reports are returned
`created_at`	Date	Date and time of report creation
`updated_at`	Date	Date and time of latest information update or report execution
`type`	Integer	Report type ID
`report_id`	String	UID, unique identifier of the report
`group`	String	Commodity group (the list of values is given in Table 4.2.1.2). ***Note:*** Available only for reports generated by methods with the `"group"` input parameter
`document_id`	String	ID of a delivery document from the Russian Federation. ***Note:*** Displayed for reports generated by the `/v2/reports/addMarkCirculation` method, if the specified parameter has been filled in
`status`	Object	Current status of the report. Class `ReportStatus` object (see cl. 4.2.5)

Example of the response body (fragment)

```
{
    "report_list": [
        {
            "created_at": "2021-05-29 13:27:55+03",
            "updated_at": "2021-05-29 13:27:56.671914+03",
            "type": 1,
            "report_id": "2249ab44-4ec2-422c-baf7-e30c05e6a4c7",
            "group": "shoes",
            "status": {
                "code": 30,
                "message": "Выполнен"
            }
        },
        {
            "created_at": "2021-05-29 13:20:35+03",
            "updated_at": "2021-05-29 13:20:36.56166+03",
            "type": 1,
            "report_id": "34127f34-7479-45fa-aa51-47ff683340f5",
            "group": "shoes",
            "status": {
                "code": 30,
```

```
                "message": "Выполнен"
            }
        },
        ...
    ]
}
```

3.5.7 "Report types" method

Request Type: `POST`
Request Line: `/v2/reports/types`
Parameters: No input parameters

Table 3.5.7.1 - HTTP response parameters, Code 200

Key	Data type	Description
`types_list`	Array of Object	Array with the list of report types. Class `ReportType` object (see cl. 4.2.8)
Example of the response body (fragment)		

```
{
    "types_list": [
        {
            "id": 1,
            "name": "Ввод в оборот"
        },
        {
            "id": 2,
            "name": "Агрегирование"
        },
        ...
        {
            "id": 4,
            "name": "Использование
        },
        ...
        {
            "id": 6,
            "name": "Ввод в оборот. Остатки"
        },
        {
            "id": 7,
            "name": "Ввод в оборот в РБ"
        },
        {
            "id": 8,
            "name": "Отмена использования"
        },
        ...
        {
            "id": 10,
            "name": "Маркировка товаров из давальческого сырья"
        },
        {
            "id": 13,
            "name": "Изменение параметров маркировки"
        }
```

```
    ]
}
```

3.5.8 "List of used codes" method

Request Type: `POST`
Request Line: `/v2/reports/labels`
Parameters:

Table 3.5.8.1 - HTTP request parameters

Key	Resp.	Data type	Description
`report_id`	Yes	String	Unique identifier of the report. UID received from the methods `/v2/reports/addMark`, `/v2/reports/addGroupMark`, `/v2/reports/addAggregate`, `/v2/reports/addUsed`, `/v2/reports/addCancelUsed`, `/v2/reports/addMarkCirculation` or `/v2/reports/addMarkLeftovers`, as well as `/v2/reports/list`
Example of the request body			

<pre>{
"report_id": "a83e9a31-d025-4927-9d6f-95d7fae5d58e"
}</pre>			

Table 3.5.8.2 - HTTP response parameters, Code 200

Key	Data type	Description
`report_id`	String	UID, unique identifier of the report
`labels`	Array of String	Array (list) of marking codes or series and numbers UCM used in the report generation
Example of the response body (fragment)		

<pre>{
"report_id": "a83e9a31-d025-4927-9d6f-95d7fae5d58e",
"labels": [
"0104810223007436212PWKhQkD 931qxa",
"0104810223007436212PWKhQk5 93RaEC",
...
"0104810223007436212PWKhQWS 932AxX"
]
}</pre>		

3.5.9 "Introduction into turnover report" method

Request Type: `POST`
Request Line: `/v2/reports/addMarkCirculation`
Parameters:

Table 3.5.9.1 – HTTP request parameters

Key	Resp.	Data type	Description
`group`	Yes	String	Commodity group (the list of values is given

			in Table 4.2.1.2)
`labels`	Yes	Array of String	Array (list) of labelling codes applied to the goods. ***Limitation:*** Max. 5,000 labelling codes per request
`circulation_reason`	Yes	String	Reason to introduce the goods into turnover. `circulation_reason` enumeration value code obtained by `/directories/{:code}` method (see 3.8.3; the list of values is given in Table 4.3.5)
`document_id`	No	String	UID, a unique identifier of a document with delivery type `type = ship`, used for acceptance, obtained from the `/ships/importList` method. ***Note:*** It is indicated if the reason `circulation_reason = import` and delivery information were sent by the exporter and received in the SIS "Electronic Mark"
`params`	No	Object	Additional parameters describing a batch of goods. ***Note:*** Mandatory if `circulation_reason = import`
`country`	Yes	String	Numerical code of a country the goods are purchased in. `country` enumeration value code obtained by `/directories/{:code}` method (see 3.8.3; the list of values is given in Table 4.3.1)
`document_date`	Yes	Date	Goods purchase document date. ***Format:*** `YYYY-MM-DD`
`document_number`	Yes	String	Goods purchase document number
`org_number`	No	String	Payer's Identification Number, Individual Taxpayer Number or other equivalent seller ID
`org_name`	No	String	Seller name

Example of the request body (fragment)

```
{
    "group": "shoes",
    "labels": [
        "0104609432871247212JvdDBvxlFsvH91TEST92testtest...testtest",
        "010460943287124721 2bpnSBlxsCSvw91TEST92testtest...testtest",
        ...
        "0104609432871247 21eSsgGsEHcwjFs91TEST92testtest...testtest"
    ],
    "circulation_reason": "import",
    "document_id": "c8303b42-fd5f-4e6d-945f-011fc6125606",
    "params":{
        "country": "643",
        "document_date": "2021-10-10",
        "document_number": "123456",
        "ogr_number": "6454122674",
        "ogr_name": "ООО \"ФРЕШ\""
    }
}
```

Table 3.5.9.2 – HTTP response parameters, Code 200

Key	Data type	Description
report_id	String	UID, unique ID number of a report
Example of the response body		

<pre>{
"report_id": "9f7a3581-9447-425a-a931-49412b3fdb19"
}</pre>		

3.5.10 "Leftovers labelling report" method

Request Type: POST
Request Line: /v2/reports/addMarkLeftovers
Parameters:

Table 3.5.10.1 – HTTP request parameters

Key	Resp.	Data type	Description
group	Yes	String	Commodity group (the list of values is given in Table 4.2.1.2). ***Note:*** Specify the commodity group subject to leftovers abelling
labels	Yes	Array	Array (list) of labelling codes applied to the goods. ***Limitation:*** Max. 5 000 labelling codes per request
label	Yes	String	Labelling code
comment	No	String	Comment, note
params	Yes	Array of Object	Additional data array describing a batch of labelled leftovers. Param class objects (see 4.1.5). ***Note:*** The only parameter used in leftovers labelling is mark_method ("Labelling method (security printing)")
Example of the request body			

<pre>{
"group": "shoes",
"labels": [
{
"label": "0104814130000030212qc7v9aauLzpW91TEST92test...test",
"comment": "Сапоги мужские, остатки"
},
{
"label": "0104609432871247215DsgGsEHcwjFs91TEST92test...test",
"comment": "Ботинки защитные, пр-во РФ"
}
]
"params": [
{
"code": "mark_method",
"value": "3"
}
]
}</pre>			

Table 3.5.10.2 – HTTP response parameters, Code 200

Key	Data type	Description
report_id	String	UID, unique identification number of a report
Example of the response body		

<pre>{
"report_id": "9f7a3581-9447-425a-a931-49412b3fdb19"
}</pre>		

Note: The method is applicable only from the date of introduction of mandatory labeling and becomes unavailable after the end of the period of transmission of information on the application of identification means to the product leftovers established for the corresponding product group.

3.5.11 "UCM application cancellation" method

Request Type: POST
Request Line: /v2/reports/addCancelUsed
Parameters:

Table 3.5.11.1 – HTTP request parameters

Key	Resp.	Data type	Description
labels	Yes	Array of String	Array (list) of UCM series and numbers eliminated from the application report. ***Limitation:*** Max. 25 000 marks per request.
Example of the request body (fragment)			

<pre>{
"labels": [
"AAA011050881",
"AAA011050882",
...
"AAA011051000"
]
}</pre>			

Table 3.5.11.2 – HTTP response parameters, Code 200

Key	Data type	Description
report_id	String	UID, unique identifier of the UCM application report
Example of the response body		

<pre>{
"report_id": "e15c7739-7e53-4df6-8f36-72bcf3c04b21"
}</pre>		

3.5.12 "Report on goods made from customer's raw materials marking" method

Request Type: POST
Request Line: /v2/reports/addMarkTrans
Parameters:

Table 3.5.12.1 - HTTP request parameters

Key	Resp.	Data type	Description
group	Yes	String	Commodity group (the list of values is given in Table 4.2.1.2).
labels	Yes	Array of String	Array (list) of marking codes applied on the goods
Example of the request body			

<pre>{
"group": "milk",
"labels": [
"0104811258000850212apa5tyq 93_dwG",
"0104811258000850212pxq5tkk 93NQw^",
...
"0104811258000850212fdkrDwxR 93dRst"
]
}</pre>			

Table 3.5.12.2 - HTTP response parameters, Code 200

Key	Data type	Description
report_id	String	UID, unique identifier of the report on marking
Example of the response body		

<pre>{
"report_id": "d9ba0436-70ee-449f-8c53-5a3624a32b9f"
}</pre>		

3.5.13 "Report on changes in marking parameters" method

Request Type: POST
Request Line: /v2/reports/edit
Parameters:

Table 3.5.13.1 - HTTP request parameters

Key	Resp.	Data type	Description
report_id	Yes	String	Unique identifier of the report being corrected. UID received from methods /v2/reports/addMark or /v2/reports/addGroupMark. ***Note:*** Available only for labeling/group labeling reports (report_type = 1)
params	Yes	Array of Objects	Array of editable parameters. Composition of the parameters (additional characteristics) and the conditions for their completing are determined by the marked goods belonging to the relevant catalog section. Class Param objects (see cl. 4.1.5 and Appendix D).
Example of the request body			

<pre>{
"group": "57fb2260-d978-4793-a674-27da1f7583d5",
"params": [
{
"code": 25,</pre>			

```
            "value": "2022-10-01"
        },
        {
            "code": 90,
            "value": "Партия 2"
        },
        {
            "code": 91,
            "value": "55 г"
        },
        {
            "code": 92,
            "value": "2022-10-20"
        },
        {
            "code": 94,
            "value": "CreamCheese"
        }
    ]
}
```


Table 3.5.12.2 - HTTP response parameters, Code 200

Key	Data type	Description
report_id	String	UID, unique identifier of the report on marking
Example of the response body		

{
"report_id": "67db7b9d-ed03-4347-a849-753eed01076c"
}		

3.6 Methods of working with counterparties

3.6.1 "Information on counterparty" method

Request Type: POST
Request Line: /contracts/agent
Parameters:

Table 3.6.1.1 - HTTP request parameters

Key	Resp.	Data type	Description
unp	Yes	String	Payer's Identification Number, Individual Taxpayer Number or other equivalent counterparty identifier
Example of the request body			

{
"unp": "700016217"
}			

Table 3.6.1.2 - HTTP response parameters, Code 200

Key	Data type	Description
	Object	Information on the counterparty. Class Agent object (see cl. 4.1.2)
Example of the response body		

```
{
    "id": 176,
    "name": "Открытое акционерное общество \"Белшина\"",
    "unp": "700016217",
    "country": "112",
    "address": "Могилевская обл., г. Бобруйск, ш. Минское, д. 4",
    "gln": "4811644900009"
}
```


3.6.2 "Addition of a new counterparty" method

Request Type: `PUT`
Request Line: `/contracts/addAgent`
Parameters:

Table 3.6.2.1 - HTTP request parameters

Key	Resp.	Data type	Description
`country`	Yes	String	Country numerical code. Enumeration value code `country` received by the method `/directories/{:code}` (see cl. 3.8.3; the list of values is given in Table 4.3.1)
`unp`	Yes	String	Payer's Identification Number, Individual Taxpayer Number or other equivalent counterparty identifier
`name`	Yes	String	Counterparty name
`address`	No	String	Counterparty address
`gln`	No	String	Counterparty GLN
Example of the request body			

{
"name": "ООО \"УНИВЕРСАЛ-ТРЕЙДИНГ\"",
"unp": "0274068089",
"address": "ул. Милашенкова, д. 4 корп. 7, г. Москва",
"country": "643",
"gln": "4630019409950"
}			

Table 3.6.2.2 - HTTP response parameters, Code 200

Key	Data type	Description
	Object	Information on the counterparty. Class `Agent` object (see cl. 4.1.2)
Example of the response body		

{
"id": 18413,
"name": "ООО \"УНИВЕРСАЛ-ТРЕЙДИНГ\"",
"unp": "0274068089",
"country": "643",
"address": "ул. Милашенкова, д. 4 корп. 7, г. Москва",
"gln": "4630019409950"
}		

3.6.3 "Counterparties list" method

Request Type: `POST`
Request Line: `/contracts/agentsList`
Parameters:

Table 3.6.3.1 - HTTP request parameters

Key	Resp.	Data type	Description
`country`	No	String	Country numerical code. Enumeration value code `country` received by the method `/directories/{:code}` (see cl. 3.8.3; the list of values is given in Table 4.3.1)
`unp`	No	String	Payer's Identification Number, Individual Taxpayer Number or other equivalent counterparty identifier
`name`	No	String	The name of the counterparty or part of it. ***Note:*** Selection is done by substring search
Example of the request body			

<pre>{
"country": "112"
}</pre>			

Table 3.6.3.2 - HTTP response parameters, Code 200

Key	Data type	Description
`agents_list`	Array of Object	Array containing information on the counterparties. Class `Agent` objects (see cl. 4.1.2). ***Note:*** The first 500 records that meet the selection criteria are returned in the response.
Example of the response body (fragment)		


```
{
    "agents_list": [
        {
            "id": 9603,
            "name": "ОАО \"Молочный Мир\"",
            "unp": "500040357",
            "country": {
                "code": "112",
                "name": "Беларусь",
                "comment": null
            },
            "address": "Республика Беларусь, г. Гродно, ул. Гаспадарчая, 28",
            "gln": "4810099900008"
        },
        {
            "id": 48,
            "name": "ОАО \"Гомельское ПО \"Кристалл\" – управляющая компания
холдинга \"КРИСТАЛЛ-ХОЛДИНГ\"",
            "unp": "400078252",
            "country": {
                "code": "112",
                "name": "Беларусь",
                "comment": null
            },
```

```
            "address": "Гомельская обл., г. Гомель,ул. Черниговская, д.22 б",
            "gln": ""
        },
        {
            "id": 49,
            "name": "Общество с ограниченной ответственность "Фабрика Чобот"",
            "unp": "191600046",
            "country": {
                "code": "112",
                "name": "Беларусь",
                "comment": null
            },
            "address": "Республика Беларусь, г.Минск Логойский тракт, д.22а",
            "gln": "4811215900001"
        },
        ...
    ]
}
```


3.7 Methods of working with shipments

3.7.1 "Codes shipment" method

Request Type: `PUT`
Request Line: `/ships/add`
Parameters:

Table 3.7.1.1 - HTTP request parameters

Key	Resp.	Data type	Description
`shipping_doc`	Yes	String	Accompanying document type. Enumeration value code `shippingDocs` received by the method `/directories/{:code}` (see cl. 3.8.3; the list of values is given in Table 4.3.3)
`nomer_tn`	Yes	String	Accompanying document number, can also contain series
`agent`	Yes	Integer	ID of the consignee counterparty to the address of which the codes are shipped and who will have to enter them. ***Note:*** The value of the attribute `id` of the class `Agent` object is specified (see cl. 4.1.2)
`count`	Yes	Integer	Total number of shipped marking or aggregation codes
`comment`	No	String	Comment to the shipment
`country`	Yes	String	Numerical code of the country to which the shipment is carried out. Enumeration value code `country` received by the method

			`/directories/{:code}` (see cl. 3.8.3; the list of values is given in Table 4.3.1)
`price`	No	Float	Total price of the goods marked with the codes.
`currency`	Yes	String	Currency of the shipment. Enumeration value code `currency` received by the method `/directories/{:code}` (see cl. 3.8.3; the list of values is given in Table 4.3.2)
`operation_date`	Yes	Date	Date of shipment (date of completion of the transaction by the participant, as a rule the date of accompanying document). ***Format:*** `YYYY-MM-DD`. ***Limitation:*** Shipment date may not be earlier than the marking code generation date
`labels`	Yes	Array of String	Array of marking codes and/or aggregation codes subject to shipment. ***Note:*** The aggregation codes are applicable to shipments within Belarus only. ***Limitations:*** The codes listed in the array must conform to the following conditions: 1. Marking codes must belong to the same commodity group; 2. The marking code must not be a part of another aggregate; 3. The aggregation code must not be "empty", i.e. it must contain nested codes (i.e. subordinate marking codes and/or low-level aggregation codes); 4. The aggregation code must not be a part of a top-level aggregate; 5. It is forbidden to simultaneously specify the marking code and the aggregation code to which it is "linked"; 6. In total, overall number of marking codes (including those included in the aggregation) per 1 shipment. 6.1. When shipped to the Russian Federation - no more

			than 10,000 codes; 6.2. In other cases - no more than 100,000 codes. 7. When shipped to the Russian Federation, aggregation codes are not transferred; only marking codes must be indicated (taking into account clause 6.1).
`eas_products`	No	Object	Additional data for the supplied goods. ***Note:*** Applicable in case of shipment data exchange in the course of transborder trade between the EAEU member states
`{:gtin}`	No	String	The array element key name corresponds to the goods GTIN, the element value includes additional shipment parameters for the specified GTIN
`product_cost`	No	String, Integer	Price per unit of goods (including VAT, if the goods are subject to VAT). *See the following note.*
`product_tax`	No	String, Integer	Amount of VAT per item. *See the following note.*
`certificate_document_data`	No	Array of Object	Data array for the documentary recognition of goods conformity to quality and/or safety regulations
`certificate_type`	Yes	String	Conformity recognition document type. `certificate_types` enumeration valure code obtained by `/directories/{:code}` (see 3.8.3; the list of values is given in Table 4.3.6)
`certificate_number`	Yes	String	Conformity recognition document number
`certificate_date`	Yes	Date	Conformity recognition document date. ***Format:*** `YYYY-MM-DD`

Example of the request body (fragment)

```
{
    "shipping_doc": "tnttn",
    "nomer_tn": "ЮЯ03456789",
    "agent": 18413,
    "count": 10,
    "comment": "Отгрузка согласно счёт-фактуре № 12456 от 04.11.2020",
    "country": "643",
    "price": 4567.89,
```

```
    "currency": "rub643",
    "operation_date": "2020-11-24",
    "labels": [
        "00048116129000000529",
        "000481901888000000320204811309016724376 1320111230987zyx",
        "0104811309016724212 8v8LZVn 936rXO",
        ...
    ],
    "eas_products": {
        "04811309016724": {
            "product_cost": "2050.75",
            "product_tax": "205.08",
            "certificate_document_data": [
                {
                    "certificate_type": "CONFORMITY_CERTIFICATE",
                    "certificate_number": "123",
                    "certificate_date": "2021-07-07"
                },
                {
                    "certificate_type": "STATE_REGISTRATION_CERTIFICATE",
                    "certificate_number": "321",
                    "certificate_date": "2021-07-07"
                }
            ]
        },
       "04810268024894": {
            "product_cost": "349.60",
            "product_tax": "0.00"
        },
        ...
    }
}
```


Table 3.7.1.2 - HTTP response parameters, Code 200

Key	Data type	Description
`message`	String	Informative message
`ship`	Integer	Shipment number (identifier)
Example of the request body		
`{ "message": "#4355 готовится к отгрузке.", "ship": 4355 }`		

Note: Additional unit price and VAT information applies if shipped to the Russian Federation and is transmitted to:

1. On a voluntary basis, including within the framework of contractual relations between the exporter (seller, shipper) and the importer (buyer, consignee).

2. In Russian rubles (RUB).

3. In one of the following formats:

Format 1. In rubles, as a number with two decimal places (signs), presented as a string without spaces in the format [0-9].[0-9]{2}, for example: `"product_cost": "123.00"` or `"product_tax": "0.00"`.

Format 2. In kopecks, as an integer, for example: `"product_cost": 12300` or `"product_tax": 0`.

3.7.2 "Information on shipment" method

Request Type: `GET`
Request Line: `/ships/{:code}`
Parameters:

Table 3.7.2.1 - HTTP request parameters

Key	Resp.	Data type	Description
`code`	Yes	Integer	Shipment number. Shipment ID received from methods `/ships/add`, `/ships/list` or `/ships/shipped`
Example of the Request Line			
`/ships/4355`			

Table 3.7.2.2 - HTTP response parameters, Code 200

Key	Data type	Description
	Object	Information on shipment. Class `Ship` object (see cl. 4.1.8)

Example of the response body


```
{
    "name": "ЮЯ03456789",
    "count": 3,
    "comment": "Отгрузка согласно счёт-фактуре № 12456 от 04.11.2020",
    "created_at": "2020-11-24 16:47:34+03",
    "real_count": 15,
    "agg_count": 2,
    "taked_at": "2020-11-24 16:47:02+03",
    "received_count": null,
    "price": {
        "price": 4567.89,
        "currency": "rub643"
    },
    "contract": false,
    "items": [
        {
            "gtin": "04810062003354",
            "name": "Молоко питьевое пастеризованное \"ВКУСНОЕ\"",
            "count": 2,
            "group": "milk",
            "articul": ""
        },
        {
            "gtin": "04810268018930",
            "name": "Кефир 2,5%",
            "count": 4,
            "group": "milk",
            "articul": ""
        },
        {
            "gtin": "04811309016724",
            "name": "Молоко питьевое",
```

```
            "count": 5,
            "group": "milk",
            "articul": null
        }
    ],
    "labels": [
        {
            "id": 12,
            "name": "Код РБ идентификации молочной продукции",
            "count": 11
        },
        {
            "id": 13,
            "name": "Код агрегации(импорт)",
            "count": 3
        }
    ],
    "operation_date": "2020-11-24",
    "shipping_doc": {
        "name": "ТН  - товарная накладная",
        "code": "tnttn"
    },
    "code": 4355,
    "status": {
        "code": 12,
        "message": "Отгружен"
    },
    "user_from": {
        "name": "СООО \"Свиток\"",
        "address": "г. Минск,ул. Тимирязева, д.2, ком. 16",
        "unp": 190704463,
        "gln": "4811612900000",
        "country": "112",
        "is_rules_agree": true
    },
    "user_to": {
        "agent": 18413,
        "name": "ООО \"УНИВЕРСАЛ-ТРЕЙДИНГ\"",
        "address": "ул. Милашенкова, д. 4 корп. 7, г. Москва",
        "unp": "0274068089",
        "gln": "4630019409950",
        "country": "643"
    },
    "country": {
        "id": "643",
        "name": "Россия"
    }
}
```


3.7.3 "List of shipped codes" method

Request Type: `POST`
Request Line: `/ships/labels`
Parameters:

Table 3.7.3.1 - HTTP request parameters

Key	Resp.	Data type	Description
code	Yes	Integer	Shipment number. Shipment ID received from methods /ships/add, /ships/list or /ships/shipped
Example of the request body			

```
{
    "code": 4355
}
```

Table 3.7.3.2 - HTTP response parameters, Code 200

Key	Data type	Description
aggregates	Array	Array of all aggregation codes (excluding nesting levels)
label	String	Aggregation Code
status	Object	Current status of the aggregation code. Class LabelStatus object (see cl. 4.2.3)
labels	Array	Array of all marking codes (excluding nesting levels)
label	String	Marking code
status	Object	Current status of the marking code. Class LabelStatus object (see cl. 4.2.3)
snomer	String	Series and number of a document with a certain protection level. ***Note:*** Used for all code types with the attribute is_bso = True
Example of the response body (fragment)		

```
{
    "aggregates": [
        {
            "label": "00048116129000000529",
            "status": {
                "code": 30,
                "message": "Валидирован"
            }
        },
        ...
        {
            "label": "000481901888000000250204811309016724376132011123O987zyx",
            "status": {
                "code": 30,
                "message": "Валидирован"
            }
        }
    ],
    "labels": [
        {
            "label": "0104810268018930212Z6rhrpa93Y!&m",
            "status": {
                "code": 50,
                "message": " Промаркирован"
            },
            "snomer": null
        },
```

```
        ...
        {
            "label": "0104810268018930212CnSujHX93BiYG",
            "status": {
                "code": 50,
                "message": "Промаркирован"
            },
            "snomer": null
        }
    ]
}
```

3.7.4 "Code entry" method

Request Type: `POST`
Request Line: `/ships/take`
Parameters:

Table 3.7.4.1 - HTTP request parameters

Key	Resp.	Data type	Description
`code`	Yes	Integer	Shipment number. Shipment ID received from the method `/ships/list`
`labels`	Yes	Array of String	Array of marking codes, aggregation codes subject to acceptance. Can be received by the method `/ships/labels`
Example of the request body			

<pre>{
"code": 4355,
"labels": [
"00048116129000000529",
"0004819018880000003202048113090167242376 1320111230987zyx",
"0104811309016724212 8v8LZVn 936rXO"
]
}</pre>			

Table 3.7.4.2 - HTTP response parameters, Code 200

Key	Data type	Description
`message`	String	Informative message
`count`	Integer	Total number of codes entered (including marking codes and aggregation codes at all nesting levels)
Example of the response body		

<pre>{
"message": "Успешно оприходовано",
"count": 15
}</pre>		

3.7.5 "List of shipments" method

Request Type: `POST`
Request Line: `/ships/shipped` - consignor context (shipment)
`/ships/list` - consignee context (acceptance)

Parameters:

Table 3.7.5.1 - HTTP request parameters

Key	Resp.	Data type	Description
`date_from`	Yes	Date	Interval start date. ***Format:*** `YYYY-MM-DD`
`date_to`	Yes	Date	Interval end date. ***Format:*** `YYYY-MM-DD`
`status`	No	Integer	Code (identifier) of the shipment status (the list of values is given in Table 4.2.6.2)
`country`	No	String	Country numerical code. Enumeration value code `country` received by the method `/directories/{:code}` (see cl. 3.8.3; the list of values is given in Table 4.3.1)
`page`	Yes	Integer	Page number of the shipment list

Example of the request body

```
{
    "date_from": "2020-11-20",
    "date_to": "2020-11-24",
    "status": 10,
    "country": "643",
    "page": 1
}
```

Table 3.7.5.2 - HTTP response parameters, Code 200

Key	Data type	Description
`count`	Integer	Total number of shipments in sampling with the current sorts (filters).
`ships_list`	Array of Object	Array containing information on shipments. Class `Ship` objects (see cl. 4.1.8)

Example of the response body

```
{
    "count": 6,
    "ships_list": [
        {
            "name": "ЮЯ03456789",
            "count": 3,
            "comment": "Отгрузка согласно счёт-фактуре № 12456 от 04.11.2020",
            "created_at": "2020-11-24 16:47:34+03",
            "real_count": 15,
            "agg_count": 2,
            "taked_at": "2020-11-24 16:47:02+03",
            "received_count": null,
            "price": {
                "price": 4567.89,
                "currency": "rub643"
            },
            "contract": false,
            "operation_date": "2020-11-24",
            "shipping_doc": {
                "name": "ТН  - товарная накладная",
                "code": "tnttn"
            },
```

```
            "code": 4355,
            "status": {
                "code": 12,
                "message": "Отгружен"
            },
            "user_from": {
                "name": "СООО \"Свиток\"",
                "address": "г. Минск,ул. Тимирязева, д.2, ком. 16",
                "unp": 190704463,
                "gln": "4811612900000",
                "country": "112",
                "is_rules_agree": true
            },
            "user_to": {
                "agent": 18413,
                "name": "ООО \"УНИВЕРСАЛ-ТРЕЙДИНГ\"",
                "address": "ул. Милашенкова, д. 4 корп. 7, г. Москва",
                "unp": "0274068089",
                "gln": "4630019409950",
                "country": "643"
            },
            "country": {
                "id": "643",
                "name": "Россия"
            }
        }
    ]
}
```


3.7.6 "Registration of dairy products supply" method

Request Type: `POST`
Request Line: `/ships/gtin`
Parameters:

Table 3.7.6.1 - HTTP request parameters

Key	Resp.	Data type	Description
`group`	Yes	String	Commodity group (the list of values is given in Table 4.2.1.2)
`shipping_doc`	Yes/No	String	Accompanying document type. Enumeration value code `shippingDocs` received by the method `/directories/{:code}` (see cl. 3.8.3; the list of values is given in Table 4.3.3). ***Note:*** Required if the type of operation is `operation = export`
`operation`	Yes	String	Type of operation: `export` - export registration; `import` - import registration
`document_id`	Yes/No	String	UID, a unique identifier of a document with delivery type

			`type = ship`, used for acceptance, obtained from the `/ships/gtinList` method. ***Note:*** Required if the type of operation is `operation = import` and delivery information were sent by the exporter and received in the SIS "Electronic Mark"
`nomer_tn`	Yes/No	String	Accompanying document number, can also contain series. ***Note:*** Required if the type of operation is `operation = export`
`country`	Yes/No	String	Numerical code of the country to which the supply is carried out. Enumeration value code `country` received by the method `/directories/{:code}` (see cl. 3.8.3; the list of values is given in Table 4.3.1). ***Notes:*** 1. The value of the attribute `id` of the class `Agent` object is specified (see cl. 4.1.2). 2. Required if the type of operation is `operation = export`
`agent`	Yes/No	Integer	ID of the consignee counterparty to the address of which the supply is carried out. ***Note:*** Required if the type of operation is `operation = export`
`comment`	No	String	Comment to the supply
`price`	No	String	Total cost of the goods supplied
`currency`	Yes/No	String	Supply currency. Enumeration value code `currency` received by the method `/directories/{:code}` (see cl. 3.8.3; the list of values is given in Table 4.3.2). ***Note:*** Required if the type of operation is `operation = export`
`operation_date`	Yes/No	Date	Date of supply (date of completion of the transaction by the participant, as a rule the date of accompanying document). ***Format:*** `YYYY-MM-DD`. ***Note:*** Required if the type of

			operation is `operation = export`
`gtin`	Yes	Array of Object	Array of volume-article accounting parameters of the supplied goods by GTIN and quantity
`{:gtin}`	Yes	String	The array element key name corresponds to the GTIN value, the value of the element contains the supply parameters for the specified GTIN. ***Template:*** [0-9]{14}
`count`	Yes	Integer	Quantity of the supplied goods. ***Limitation:*** If the type of operation is `operation = export`, then no more than 500,000 per 1 GTIN
`certificate`	No	String	Globally Unique Identifier (GUID) of the electronic veterinary accompanying document in SIS "AITS"
`name`	No	String	Goods name
`group`	Yes	String	Commodity group (the list of values is given in Table 4.2.1.2)
`product_cost`	No	String, Integer	Price per unit of goods (including VAT, if the goods are subject to VAT). *See the following note 2.*
`product_tax`	No	String, Integer	Amount of VAT per item. *See the following note 2.*
`certificate_document_data`	No	Array of Object	Data array on the documentary proof of product qualification as to quality and/or safety
`certificate_type`	Yes	String	Type of the qualification document. `certificate_types` listing value code obtained by `/directories/{:code}` method (see 3.8.3; list of values in table 4.3.6)
`certificate_number`	Yes	String	Number of the qualification document
`certificate_date`	Yes	Date	Date of the qualification document. ***Format:*** `YYYY-MM-DD`
Example of the request body (fragment)			
Option 1. Export registration			

```
{
    "group": "milk",
    "shipping_doc": "tnttn",
    "operation": "export",
    "nomer_tn": "GJ0446918",
    "country": "643",
```

```
    "agent": 18774,
    "comment": "Поставка по заказу N 9848 от 14.05.2021",
    "price": "23456.78",
    "currency": "rub643",
    "operation_date": "2021-05-28",
    "gtin": {
        "04810263033914": {
            "count": 10,
            "certificate": "a51ccbdf-eb38-44e2-9234-1ad1ad06f502",
            "name": "Молоко 3,2%",
            "product_cost": "2050.75",
            "product_tax": "205.08",
            "certificate_document_data": [
                {
                    "certificate_type": "CONFORMITY_CERTIFICATE",
                    "certificate_number": "123",
                    "certificate_date": "2021-07-07"
                },
                {
                    "certificate_type": "STATE_REGISTRATION_CERTIFICATE",
                    "certificate_number": "321",
                    "certificate_date": "2021-07-07"
                }
            ]
        },
        "04810767009354": {
            "count": 70,
            "certificate": "a51ccbdf-eb38-44e2-9234-1ad1ad06f502",
            "name": "Молоко обезжиренное"
        },
        ...
    }
}
```


Option 2. Import registration


```
{
    "group": "milk",
    "operation": "import",
    "document_id": "702bb042-f8d1-4037-g6r1-7febe1043b3f",
    "gtin": {
        "04607161623930": {
            "count": 48
        },
        "04810898002477": {
            "count": 152
        },
        ...
    },
    "comment": ""
}
```


Table 3.7.6.2 - HTTP response parameters, Code 200

Key	Data type	Description
	Object	Information on the supply. Class `Ship` object (see cl. 4.1.8)
Example of the response body		


```
{
```

```
    "created_at": "2021-05-30 10:31:45",
    "name": "ГЖ0446918",
    "comment": "Поставка по заказу N 9848 от 14.05.2021",
    "price": {
        "price": "234.56",
        "currency": "rub643"
    },
    "contract": false,
    "operation_date": "2021-05-28",
    "shipping_doc": {
        "name": "ТН  - товарная накладная",
        "code": "tnttn"
    },
    "code": "498662",
    "user_from": {
        "name": "СООО \"Свиток\"",
        "address": "г. Минск,ул. Тимирязева, д.2, ком. 16",
        "unp": 190704463,
        "gln": "4811612900000",
        "country": "112",
        "is_rules_agree": true
    },
    "user_to": {
        "agent": 18774,
        "name": "АО \"Тандер\"",
        "unp": "350002",
        "country": "643",
        "address": "350002 г.Краснодар, ул.Леваневского, д.185",
        "gln": ""
    },
    "country": {
        "id": "643",
        "name": "Россия"
    },
    "gtin": {
        "04810263033914": {
            "count": 10,
            "certificate": "a51ccbdf-eb38-44e2-9234-1ad1ad06f502"
        },
        "04810767009354": {
            "count": 70,
            "certificate": "a51ccbdf-eb38-44e2-9234-1ad1ad06f502"
        },
        ...
    }
}
```


Notes:

1. The method is only used to transfer information on the supply (shipment) of goods from the "Dairy products" group to the Russian Federation as part of the volume-article accounting, as well as imports from the Russian Federation.

2. Additional unit price and VAT information applies if shipped to the Russian Federation and is transmitted to:

1. On a voluntary basis, including within the framework of contractual relations between the exporter (seller, shipper) and the importer (buyer, consignee).

2. In Russian rubles (RUB).

3. In one of the following formats:

Format 1. In rubles, as a number with two decimal places (signs), presented as a string without spaces in the format [0-9].[0-9]{2}, for example: `"product_cost": "123.00"` or `"product_tax": "0.00"`.

Format 2. In kopecks, as an integer, for example: `"product_cost": 12300` or `"product_tax": 0`.

3.7.7 "List of dairy products supply" method

Request Type: `POST`
Request Line: `/ships/gtinList`
Parameters:

Table 3.7.7.1 - HTTP request parameters

Key	Resp.	Data type	Description
`date_from`	Yes	Date	Interval start date. ***Format:*** `YYYY-MM-DD`
`date_to`	Yes	Date	Interval end date. ***Format:*** `YYYY-MM-DD`
`operation`	Yes	String	Type of operation: `export` - export registration; `import` - import registration
`page`	Yes	Integer	Page number of the supply list

Example of the request body

```
{
    "date_from": "2021-05-28",
    "date_to": "2021-05-28",
    "operation": "export",
    "page": 1
}
```

Table 3.7.7.2 - HTTP response parameters, Code 200

Key	Data type	Description
`count`	Integer	Total number of supplies in sampling with the current sorts (filters).
`ships_list`	Array of Object	Array containing information on the supplies. Class `Ship` objects (see cl. 4.1.8)

Example of the response body (fragment)

```
{
    "count": 6,
    "ships_list": [
        {
            "created_at": "2021-05-28 16:07:14+03",
            "name": "НД0294125414",
            "comment": "Поставка на выставку-ярмарку",
            "price": {
                "price": "234.56",
                "currency": "rub643"
            },
            "contract": false,
            "operation_date": "2021-05-28",
```

```
            "shipping_doc": {
                "name": "ТН  - товарная накладная",
                "code": "tnttn"
            },
            "code": 498655,
            "user_from": {
                "name": "СООО \"Свиток\"",
                "address": "г. Минск,ул. Тимирязева, д.2, ком. 16",
                "unp": 190704463,
                "gln": "4811612900000",
                "country": "112",
                "is_rules_agree": true
            },
            "user_to": {
                "agent": 18774,
                "name": "АО \"Тандер\"",
                "unp": "350002",
                "country": "643",
                "address": "350002 г.Краснодар, ул.Леваневского, д.185",
                "gln": ""
            },
            "country": {
                "id": "643",
                "name": "Россия"
            },
            "gtin": {
                "04810005000853": {
                    "count": 8,
                    "certificate": "e5df5f0e-724e-4169-b5c2-426983a5eafe"
                }
            }
        },
        ...
    ]
}
```


3.7.8 "List of supplies from the Russian Federation" method

Request Type: `POST`
Request Line: `/ships/importList`
Parameters:

Table 3.7.8.1 – HTTP request parameters

Key	Resp.	Data type	Description
`date_from`	Yes	Date	Period starting date. ***Format:*** `YYYY-MM-DD`
`date_to`	Yes	Date	Period end date. ***Format:*** `YYYY-MM-DD`
`page`	Yes	Integer	List of shipments page number
Example of the request body			
{ "date_from": "2021-11-01", "date_to": "2021-11-30", "page": 1			

```
}
```

Table 3.7.8.2 – HTTP response parameters, Code 200

Key	Data type	Description
count	Integer	Total number of shipments in the selection with the current criteria (filters)
shipments_list	Array of Object	Array of shipment data. Objects of Import class (see clause 4.1.12)

Example of the response body (fragment)

```
{
    "count": 6,
    "shipments_list": [
        {
            "document_id": "1ed99575-8f9d-4470-bfe4-e39be9ab5299",
            "exporter_unp": "9715225506",
            "exporter_name": "ООО ТрейдЛогСервис",
            "group": "shoes",
            "type": "manual",
            "document_date": "2021-11-23",
            "document_number": "LO00980S",
            "status": {
                "code": "10",
                "message": "Обработан"
            }
        },
        {
            "document_id": "f1s13r1z-9sh4-9dRs-gd1x-gc3x69ay3215",
            "exporter_unp": "9715225506",
            "exporter_name": "ООО ТрейдЛогСервис",
            "group": "shoes",
            "type": "export",
            "document_processing_date": "2021-11-23",
            "status": {
                "code": "10",
                "message": "Обработан"
            }
        },
        {
            "document_id": "14979997-d854-4215-820f-65862102479f",
            "exporter_unp": "9715225506",
            "exporter_name": "ООО ТрейдЛогСервис",
            "group": "shoes",
            "type": "ship",
            "document_processing_date": "2021-11-23",
            "document_number": "5",
            "document_date": "2021-11-18",
            "sender_country_code": "643",
            "status": {
                "code": 10,
                "message": "Обработан"
            }
        },
        ...
    ]
}
```

3.7.9 "Acceptance of supply from the Russian Federation" method

Request Type: `POST`
Request Line: `/ships/addImport`
Parameters:

Table 3.7.9.1 – HTTP request parameters

Key	Resp.	Data type	Description
`group`	Yes	String	Product group (see the values listed in Table 4.2.1.2)
`document_id`	Yes	String	UID, a unique identifier of a document with delivery type is `type = ship` or `type = export`, that is used to carry out acceptance and which is obtained from the method `/ships/importList`. ***Note***: Specified when the following conditions are met simultaneously: 1. Information about the delivery was transferred by the exporter and received in the SIS "Electronic Mark"; 2. The current status of the delivery document is "Processed" (`status.code = 10`); 3. It is planned to carry out a delayed or partial introduction of marking codes into turnover, including step by step introduction.
`document_date`	Yes/No	Date	Date of the goods purchase confirmation document. ***Format:*** `YYYY-MM-DD` ***Note***: Required if the delivery document, upon acceptance, has type `type = export`
`document_number`	Yes/No	String	Number of the goods purchase confirmation document. ***Note***: Required if the delivery document, upon acceptance, has type `type = export`
`exporter_unp`	Yes/No	String	Russian seller (consigner) taxpayer's number (INN). ***Note***: Required if the delivery document, upon acceptance, has type `type = export`
`exporter_name`	Yes/No	String	Name of the seller (consignor) of the Russian Federation. ***Note***: Required if the delivery document, upon acceptance, has type `type = export`
`labels`	Yes	Array of String	An array of identification codes accepted for accounting received from a consigner from the Russian Federation. ***Note***: If `document_id` is filled in, then there is a list of only identification codes from the specified delivery that are actually taken into account.
Example of the request body (fragment)			

Option 1. Trusted acceptance based on the register of deliveries or other information available to the importer

```
{
    "exporter_unp": "9715225506",
    "exporter_name": "ООО ТрейдЛогСервис",
    "group": "shoes",
    "document_date": "2021-11-23",
    "document_number": "LO00980S",
    "labels": [
        "010468036426613521AWO-gnMkmL.nQ",
        "010463009450755821EngFpOSW2ln5U",
        " 010463009450711421 4PDZkODZ>=Wyl",
        ...
        "010463009450740421eC!uBoXfj*ZWF"
     ]
}
```

Option 2: Trusted acceptance based on a cross-border shipping document handed over by the exporter and received by the importer

```
{
    "group": "shoes",
    "document_id": "14979997-d854-4215-820f-65862102479f",
    "labels": [
        "0104630094507114215 4Q0mQgbbhy2f",
        "010463009450711421 5pONkFIgI3JJM",
        "010463009450711421 5fHGdolEgjkvR",
        ...
        "010463009450711421 5NvFUN+,%upsh"
     ]
}
```

Table 3.7.9.2 – HTTP response parameters, Code 200

Key	Data type	Description
	Object	Supply information. Objects of `Import` class (see clause 4.2.12)
Example of the response body		

```
{
    "document_id": "1ed99575-8f9d-4470-bfe4-e39be9ab5299",
    "exporter_unp": "9715225506",
    "exporter_name": "ООО ТрейдЛогСервис",
    "group": "shoes",
    "type": "manual",
    "document_date": "2021-11-23",
    "document_number": "LO00980S",
    "status": {
        "code": "10",
        "message": "Обработан"
    }
}
```

3.7.10 "Information about the supply from the Russian Federation" method

Request Type: `POST`
Request Line: `/ships/import`
Parameters:

Table 3.7.10.1 – HTTP request parameters

Key	Resp.	Data type	Description
`document_id`	Yes	String	UID, unique identifier of the shipment obtained by `/ships/importList` or `/ships/addImport` method
`details`	No	Boolean	Indicator for displaying detailed information about the processing result for each identification code: `False` – information is not displayed; `True` – information is displayed. ***Note:*** By default `details = false`. Available for deliveries with the current status "Credited" (`status = 30`)
Example of the request body			
Option 1. Default			

<pre>{
"document_id": "14979997-d854-4215-820f-65862102479f"
}</pre>			

Option 2. Displaying detailed information about the processing result for each identification code (`details = true`)			

<pre>{
"document_id": "14979997-d854-4215-820f-65862102479f",
"details": true
}</pre>			

Table 3.7.10.2 – HTTP response parameters, Code 200

Key	Data type	Description
	Object	Supply information. Objects of `Import` class (see clause 4.2.12)
Example of the response body (fragment)		
Option 1. Default		


```
{
    "document_id": "14979997-d854-4215-820f-65862102479f",
    "exporter_unp": "9715225506",
    "exporter_name": "ООО ТрейдЛогСервис",
    "group": "shoes",
    "type": "ship",
    "created_at": "2021-11-23 10:40:35+03",
    "user_to": {
        "name": "СООО \"Свиток\"",
        "address": "г. Минск,ул. Тимирязева, д.2, ком. 16",
        "unp": "190704463",
        "gln": "4811612900000",
        "country": "112",
        "is_rules_agree": true
    },
    "document_processing_date": "2021-11-23",
    "document_number": "5",
    "document_date": "2021-11-18",
    "sender_country_code": "643",
    "info": {
        "received": 22,
        "not_received": 8
    },
```

```
    "status": {
        "code": "30",
        "message": "Оприходован"
    },
    "labels": {
        "0104630047061977 21HkR,&b16+_m38",
        "0104690244973442 21<HP9iSyVp47Jk",
        "0104630094507558 21EngFpOSW2ln5U",
        ...
        "0104690244973473 21yS&Awke)RpI2O"
    ]
}
```


Option 2. Displaying detailed information about the processing result for each identification code (`details = true`)


```
{
    "document_id": "14979997-d854-4215-820f-65862102479f",
    "exporter_unp": "9715225506",
    "exporter_name": "ООО ТрейдЛогСервис",
    "group": "shoes",
    "type": "ship",
    "created_at": "2021-11-23 10:40:35+03",
    "user_to": {
        "name": "СООО \"Свиток\"",
        "address": "г. Минск,ул. Тимирязева, д.2, ком. 16",
        "unp": "190704463",
        "gln": "4811612900000",
        "country": "112",
        "is_rules_agree": true
    },
    "document_processing_date": "2021-11-23",
    "document_number": "5",
    "document_date": "2021-11-18",
    "sender_country_code": "643",
    "info": {
        "received": 22,
        "not_received": 8
    },
    "status": {
        "code": "30",
        "message": "Оприходован"
    },
    "labels": {
        {
            "label": "0104630047061977 21HkR,&b16+_m38",
            "status": 1
        },
        {
            "label": "0104690244973442 21<HP9iSyVp47Jk",
            "status": 0
        },
        {
            "label": "0104630094507558 21EngFpOSW2ln5U",
            "status": 1
        },
        ...
        {
            "label": "0104690244973473 21yS&Awke)RpI2O",
```

```
            "status": 0
        }
    ]
}
```

3.7.11 "Information about the dairy products supply" method

Request Type: `POST`
Request Line: `/ships/getGtin`
Parameters:

Table 3.7.9.1 – HTTP request parameters

Key	Resp.	Data type	Description
Option 1. The supply of dairy products for export (operation = export)			
`code`	Yes	String	Supply number. Shipment ID received from the method `/ships/add`, `/ships/gtin` or `/ships/gtinList`
Option 2. The supply of dairy products by import (`operation = import`)			
`document_id`	Yes	String	UID, a unique identifier of a dairy products supply document obtained by `/ships/gtinList` method
Example of the request body (fragment)			
Option 1. The supply of dairy products for export (`operation = export`)			

<pre>{
"code": "499524"
}</pre>			

Option 2. The supply of dairy products by import (`operation = import`)			

<pre>{
"document_id": "4d421e32-bd61-4db8-badc-d3fe37755d1d"
}</pre>			

Table 3.7.9.2 – HTTP response parameters, Code 200

Key	Data type	Description
	Object	Information about the supply of dairy products. Objects of `Ship` class (see clause 4.1.8)
Example of the response body		
Option 1. The supply of dairy products for export (`operation = export`)		

```
{
    "created_at": "2022-10-10 14:55:38+03",
    "name": "004-4811612900000-00000000000000000123456789",
    "comment": "etn",
    "price": {
        "price": "123456.78",
        "currency": "rub643"
    },
    "contract": false,
    "operation_date": "2022-10-10",
    "shipping_doc": {
        "code": "tnttn",
        "name": "ЭТН – электронная товарная накладная"
    },
    "code": 499524,
```

```
    "user_from": {
        "name": "СООО \"Свиток\"",
        "address": "г. Минск,ул. Тимирязева, д.2, ком. 16",
        "unp": "190704463",
        "gln": "4811612900000",
        "country": "112",
        "is_rules_agree": true
    },
    "user_to": {
        "name": "АО \"Тандер\"",
        "unp": "350002",
        "country": "643",
        "address": "350002 г.Краснодар, ул.Леваневского, д.185",
        "gln": "",
        "agent": 18774,
        "is_verified": false
    },
    "country": {
        "id": "643",
        "name": "Россия"
    },
    "gtin": {
        "04810099033898": {
            "name": "Напиток молочный",
            "count": 18,
            "certificate": "123e4567-e89b-12d3-a456-426655440000",
            "product_tax": "2223",
            "product_cost": "4369",
            "certificate_document_data": [
                {
                    "certificate_date": "2021-11-08",
                    "certificate_type": "CONFORMITY_CERTIFICATE",
                    "certificate_number": "12"
                }
            ]
        },
        "04810268046698": {
            "name": "Сливки",
            "count": 152,
            "certificate": "a51ccbdf-eb38-44e2-9234-1ad1ad06f502"
        },
        ...
        "04813494064351": {
            "name": "Творог",
            "count": 15,
            "certificate": "a51ccbdf-eb38-44e2-9234-1ad1ad06f502"
        }
    },
    "document_id": "UEGCI000000000000000015-0011",
    "shipment_status": {
        "code": "CHECKED_NOT_OK",
        "message": "Обработан с ошибками"
    },
    "debug_message": "03: Недопустимый формат значения поля \"ИНН получателя в
РФ\"., 07: Недопустимое количество символов в значении поля \"ИНН получателя в
РФ\"., 05: Значение поля \"Сумма НДС\" находится вне допустимого числового
диапазона."
```

```
}
```

Option 2. The supply of dairy products by import (`operation = import`)


```
{
    "created_at": "2022-11-09 10:09:17+03",
    "name": "0123456",
    "comment": null,
    "price": {
        "price": null,
        "currency": "rub643"
    },
    "contract": false,
    "operation_date": "2022-11-09",
    "shipping_doc": {
        "code": "tnttn",
        "name": "ТН – товарная накладная"
    },
    "code": 499549,
    "user_from": {
        "name": " АО \"Тандер\"",
        "unp": "2310031475",
        "country": "643",
        "address": "350002 г.Краснодар, ул.Леваневского, д.185",
        "gln": "",
        "agent": 18774,
        "is_verified": false
    },
    "user_to": {
        "name": "СООО \"Свиток\"",
        "address": "г. Минск,ул. Тимирязева, д.2, ком. 16",
        "unp": "190704463",
        "gln": "4811612900000",
        "country": "112",
        "is_rules_agree": true
    },
    "country": {
        "id": "643",
        "name": "Россия"
    },
    "gtin": {
        "04648349498834": {
            "count": 5
        },
        "04690228010347": {
            "count": 3
        }
    },
    "document_id": "4d421e32-bd61-4db8-badc-d3fe37755d1d",
    "accepted": {
        "04648349498834": {
            "name": "сгущенка",
            "count": 5,
            "certificate": "153d164d-9808-4fdz-b721-d92962t46236"
        },
        "04690228010347": {
            "name": "молоко питьевое цельное пастеризованное",
            "count": 3,
            "certificate": "a785hca2-3934-8b03-a7e1-62ba0efcceee"
```

```
        }
    },
    "shipment_status": {
        "code": "ACCEPTED",
        "message": "Принят"
    }
}
```

3.8 Supplementary methods

3.8.1 "File Download" method

Request Type: `GET`
Request Line: `/downloads/{:filename}`
Parameters:

Table 3.8.1.1 - HTTP request parameters

Key	Resp.	Data type	Description
`filename`	Yes	String	File name. As a rule, specified in the similarly-named parameter of the section `file` of the class `order` object returned by the methods `/orders/add` or `/orders/{:id}`
Example of the Request Line			
`/downloads/507-sygn-253590-2020-11-19-14-01-01-4.txt`			

3.8.2 "List of enumeration guides" method

Request Type: `POST`
Request Line: `/directories/`
Parameters:

Table 3.8.2.1 - HTTP response parameters, Code 200

Key	Data type	Description
`directories`	Array of Object	Array containing information on the enumeration guides. Class `Directory` object (see cl. 4.1.9)
Example of the response body (fragment)		

```
{
    "directories": [
        {
            "code": "country",
            "name": "Страны",
            "comment": "",
            "count": 251
        },
        {
            "code": "currency",
            "name": "Валюты",
            "comment": "",
            "count": 5
        },
```

```
        {
            "code": "shippingDocs",
            "name": "Тип документов",
            "comment": "Документы прилагаемые к ...",
            "count": 16
        },
        ...
    ]
}
```

3.8.3 "Elements of enumeration guide" method

Request Type: `POST`
Request Line: `/directories/{:code}`
Parameters:

Table 3.8.3.1 - HTTP request parameters

Key	Resp.	Data type	Description
`code`	Yes	String	List code received by the method `/directories/` (see cl. 3.8.2)
Example of the Request Line			
`/directories/currency`			

Table 3.8.3.2 - HTTP response parameters, Code 200

Key	Data type	Description
`inner`	Array of Object	Array of guide elements (list, enumeration), attribute of a class object `Directory` (see cl. 4.1.9)
Example of the response body		

```
{
    "code": "currency",
    "name": "Валюты",
    "comment": "",
    "inner": [
        {
            "code": "usd840",
            "name": "Доллар США",
            "comment": null
        },
        {
            "code": "rub643",
            "name": "Российский рубль",
            "comment": null
        },
        {
            "code": "kzt398",
            "name": "Казахский тенге",
            "comment": null
        },
        {
            "code": "byn933",
            "name": "Белорусский рубль",
            "comment": null
        },
```

```
        {
            "code": "eur978",
            "name": "Евро",
            "comment": null
        }
    ]
}
```


Note: Values of some enumeration guides used as parameter values of the individual API methods are given in cl. 4.3 of this document.

3.8.4 "List of printing houses" method

Request Type: `POST`
Request Line: `/users/typographyList`
Parameters: No input parameters

Table 3.8.4.1 - HTTP response parameters, Code 200

Key	Data type	Description
`typograpy_list`	Array of Object	Array with the list of printing houses. Class `Agent` object (see cl. 4.1.2)
Example of the response body (fragment)		


```
{
    "typography_list": [
        {
            "id": 2643,
            "name": "ООО \"Типография\"",
            "address": "г. Минск",
            "unp": "101101101",
            "gln": "4810253691253",
            "country": {
                "code": "112",
                "name": "Беларусь",
                "comment": null
            }
        },
        {
            "id": 1246,
            "name": "Tetra Pak",
            "address": "Сингапур",
            "unp": "",
            "gln": null,
            "country": {
                "code": "702",
                "name": "Сингапур",
                "comment": null
            }
        },
        ...
    ]
}
```

3.9 Methods of working with requests

3.9.1 "Request for the manufacture of packaging in a printing house" method

Request Type: `POST`
Request Line: `/requests/addTypographyOrder`
Parameters:

Table 3.9.1.1 - HTTP request parameters

Key	Resp.	Data type	Description
`group`	Yes	String	Product group. ***Note:*** Available only for the product group "Dairy products" (`group = milk`)
`gtin`	Yes	String	GTIN of the goods. ***Template:*** [0-9]{14}
`count`	Yes	Integer	Number of codes ordered. ***Limitation***: Not more than 100,000 codes per 1 request
`typography_id`	Yes	Integer	The printing house ID obtained by the `/users/typographylist method` (see section 3.8.4)
`comment`	No	String	Comment to the request

Example of the request body

```
{
    "group": "milk",
    "gtin": "04812656001012",
    "count": 32000,
    "typography_id": 2673,
    "comment": "Вариант упаковки по согласованию"
}
```

Table 3.9.1.2 - HTTP response parameters, Code 200

Key	Data type	Description
`typography_order`	Object	Request for the manufacture of packaging. Object of `TypographyOrder` class (see section 4.1.13)

Example of the response body

```
{
    "typography_order": {
        "id": "126",
        "created_at": "2022-05-26 19:28:49",
        "group": "milk",
        "quantity": 32000,
        "document_id": "da99235d-d373-4f4c-80a7-72f48901299d",
        "comment": "Вариант упаковки по согласованию",
        "typography": {
            "id": 2673,
            "name": "СПТ-Типография",
            "address": "г. Москва",
            "unp": "1234567891",
            "gln": "4810002580001"
        },
        "item": {
```

```
            "name": "Масло сладкосливочное несолёное",
            "articul": null,
            "gtin": "04812656001012",
            "group": "milk",
            "document_id": null,
            "is_my": true
        },
        "status": {
            "code": 20,
            "message": "Подтвержден"
        },
        "user": {
            "name": "Валерий",
            "lastname": "Матоха",
            "info": {
                "name": "СООО \"Свиток\"",
                "address": "г. Минск,ул. Тимирязева, д.2, ком. 16",
                "unp": "190704463",
                "gln": "4811612900000",
                "country": "112",
                "is_rules_agree": true
            }
        }
    }
}
```


3.9.2 "List of requests for the manufacture of packaging in a printing house" method

Request Type: `POST`
Request Line: `/requests/listTypographyOrders`
Parameters:

Table 3.9.2.1 - HTTP request parameters

Key	Resp.	Data type	Description
`date_from`	No	Date	Interval start date. ***Format:*** `YYYY-MM-DD`
`date_to`	No	Date	Interval end date. ***Format:*** `YYYY-MM-DD`
`status`	No	Integer	Code (identifier) of the request status (the list of values is given in Table 4.2.10.2)
`page`	No	Integer	Page number of the requests list
Example of the request body			

{
"date_from": "2022-05-01",
"date_to": "2022-05-26",
"status": 20,
"page": 1
}			

Table 3.9.2.2 - HTTP response parameters, Code 200

Key	Data type	Description
`count`	Integer	Total number of requests in sampling w1ith

		the current sorts (filters).
typography_order_list	Array of Object	Array containing information on requests. Class TypographyOrder objects (see section 4.1.13)

Example of the response body


```
{
    "count": 31,
    "typography_order_list": [
        {
            "id": 126,
            "created_at": "2022-05-26 19:28:49+03",
            "group": "milk",
            "quantity": 32000,
            "document_id": "da99235d-d373-4f4c-80a7-72f48901299d",
            "comment": "Вариант упаковки по согласованию",
            "typography": {
                "id": 2673,
                "name": "СПТ-Типография",
                "address": "г. Москва",
                "unp": "1234567891",
                "gln": "4810002580001"
            },
            "item": {
                "name": "Масло сладкосливочное несолёное",
                "articul": null,
                "gtin": "04812656001012",
                "group": "milk",
                "document_id": null,
                "is_my": true
            },
            "status": {
                "code": 20,
                "message": "Подтвержден"
            },
            "user": {
                "name": "Валерий",
                "lastname": "Матоха",
                "info": {
                    "name": "СООО \"Свиток\"",
                    "address": "г. Минск,ул. Тимирязева, д.2, ком. 16",
                    "unp": "190704463",
                    "gln": "4811612900000",
                    "country": "112",
                    "is_rules_agree": true
                }
            }
        },
        ...
        {
            "id": 38,
            "created_at": "2022-05-23 14:41:44+03",
            "group": "milk",
            "quantity": 40,
            "document_id": "3b1816e9-42af-4423-9b00-da535473e202",
            "comment": "",
            "typography": {
```

```
                "id": 2673,
                "name": "СПТ-Типография",
                "address": "г. Москва",
                "unp": "1234567891",
                "gln": "4810002580001"
            },
            "item": {
                "name": "Йогурт,",
                "articul": null,
                "gtin": "04810268024375",
                "group": "milk",
                "document_id": null,
                "is_my": false
            },
            "status": {
                "code": 20,
                "message": "Подтвержден"
            },
            "user": {
                "name": "Иван",
                "lastname": "Иванов",
                "info": {
                    "name": "СООО \"Свиток\"",
                    "address": "г. Минск,ул. Тимирязева, д.2, ком. 16",
                    "unp": "190704463",
                    "gln": "4811612900000",
                    "country": "112",
                    "is_rules_agree": true
                }
            }
        }
    ]
}
```


4 OBJECTS, GUIDES AND ENUMERATIONS

The listing of object parameters, description of their structure and their potential values in this section 4 are given for reference and may be amended or changed in the higher API versions. The output of certain object parameters in an HTTP response depends on the API method the object is addressed by.

4.1 Classes of objects

4.1.1 "User" object

Class name: `User`
Parameters:

Table 4.1.1 - Structure of class metadata `User`

Key	Data type	Description
`id`	Integer	User code (identifier)
`name`	String	Own name
`lastname`	String	Surname
`email`	String	Email address

Key	Data type	Description
status	Integer	Current status: 0 - locked; 1 - active
expires_in	Date	The expiration date and time of the authorization token. ***Note:*** Used in the /auth authorization method (see cl. 3.1.1)
role	Object	Participant's role. Class UsersRoles object (see cl. 4.2.7)
info_id	Integer	Subject (participant) ID
need_change_password	Boolean	Indicator that the password needs to be changed: False – not required; True - required
info	Object	Information on the counterparty (participant). Class Agent object (see cl. 4.1.2)

4.1.2 "Counterparty" object

Class name: Agent
Parameters:

Table 4.1.2 - Structure of class metadata Agent

Key	Data type	Description
id	Integer	Counterparty code (identifier). ***Note:*** In the Ship class object parameter user_to , the specified attribute is named as "agent"
name	String	Counterparty name
addres	String	Counterparty address
unp	String	Payer's Identification Number, Individual Taxpayer Number or other equivalent counterparty identifier
gln	String	Counterparty GLN
country	String, Object	Counterparty's registration country. Enumeration value code country received by the method /directories/{:code} (see cl. 3.8.3; the list of values is given in Table 4.3.1)
is_rules_agree	Boolean	Indicator of turnover participant’s familiarisation and acceptance of the terms of the standard form contract with the Operator for the implementation of identification means and services for the provision and accounting of marking codes in the SIS "Electronic Mark" (the fact of accession to the standard form contract): True - yes; False – no.
Is_verified	Boolean	Indicator of a participant in the turnover of

Key	Data type	Description
		goods registered in the SIS "Electronic Mark": `True` – is a participant in the turnover of goods; `False` – is not a participant in the turnover of goods.

4.1.3 "Goods catalog section (category)" object

Class name: `Category`
Parameters:

Table 4.1.3.1 - Structure of class metadata `Category`

Key	Data type	Description
`code`	Integer	Catalog section code (identifier)
`name`	String	Catalog section name
`group`	Object	Information on the commodity group. Class `Group` object (see cl. 4.2.1)
`inner`	Array of Object	Array of subordinate (low-level, nesting) catalog sections. Class objects `Category`
`catalog_params`	Array of Object	Array (set, composition) of additional parameters assigned to a catalog section describing the goods integrally
`group`	String	Parameters group name
`params`	Array of Object	Set of parameters and characteristics. Class `Param` objects (see cl. 4.1.5)
`mark_params`	Array of Object	Array (set, composition) of additional parameters assigned to a catalog section describing a lot of the marked goods
`group`	String	Parameters group name
`params`	Array of Object	Set of parameters and characteristics. Class `Param` objects (see cl. 4.1.5)
`items`	Array of Object	Array containing the descriptions of goods in the specified catalog section entered by the participant. Class `Item` objects (see cl. 4.1.4)

Values of the "Catalog sections" directory are listed in Table 4.1.3.2.

Table 4.1.3.2 - List of goods catalog sections

code / **Code**	**name** / **Name**
10002	Shoes
10081	Consumer industry
10082	Bed cloths, table linen, toilet and kitchen linen
10083	Woven tunics, fleeces and blousons
10084	Overcoats, short overcoats, capes, raincoats, coats (including ski coats), wind jackets, wind breakers and similar articles
10085	Articles of clothing made of real or composition leather
10086	Tires and treads

code / Code	name / Name
10087	Perfumery
10088	Cameras and flash lamps
10089	Dairy products
10090	Goods subject to UCM marking
10092	Leftovers. Footwear products
10093	Leftovers. Tyres
10094	Packaged water
10095	Leftovers. Consumer goods

Note: Catalogue section identifiers (codes) are different in the commercial and test zones. Those in the table are commercial zone catalogue section codes.

4.1.4 "Goods" object

Class name: `Item`
Parameters:

Table 4.1.4.1 - Structure of class metadata `Item`

Key	Data type	Description
gtin	String	Product GTIN
name	String	Goods name
For the goods registered in the catalogue		
image	String	URL link to the goods image
articul	String	Goods article
code	String	Internal code (identifier) of the goods information record
description	String	Goods description
gtin_check	Boolean	Indicator of GTIN registration in the catalog: `True` - goods registered; `False` - goods registration pending.
group	String	Commodity group (the list of values is given in Table 4.2.1.2)
debug_message	Object	Service message
is_my	Boolean	Indicator of own goods: `True` - goods added by the current participant; `False` – goods added by another participant.
catalog	Object	Information on catalog section. Class `Category` object (see cl. 4.1.3)
params	Array of Object	Array of additional parameters describing the goods. Composition of the parameters (additional characteristics) and the conditions for their completing are determined by the goods belonging to the relevant catalog section. Class `Param` objects (see cl. 4.1.5)
For imported goods		
unp	String	Payer's Identification Number, Individual Taxpayer Number or other equivalent goods record owner ID
tnved	String	EAEU HS Code

Key	Data type	Description
brand	String	Brand, trademark
country	Object	Country of goods origin, country enumeration value obtained by /directories/{:code} method (see 3.7.3; the list of values is given in Table 4.3.1)
is_import	Boolean	Attribute of the imported goods added by the national operators of the EAEU member states

The values of the "Goods" directory related to the "Goods subject to UCM labeling" product group (group = ukz) are listed in Table 4.1.4.2.

Table 4.1.4.2 - List of sections of goods subject to UCM labeling

code	name
100900000695388	Caviar; canned fish (EAEU HS from 1604)
100900000695389	Water (EAEU HS from 2202)
100900000695390	Watches (EAEU HS from 9101, from 9102, from 9105)
100900000695391	TVs (EAEU HS from 8528)
100900000695392	Monitors (EAEU HS from 8528)
100900000695393	Physical data carriers (EAEU HS from 8523)
100900000695394	Mobile phones in consumer packaging, except for used ones (EAEU HS from 8517 13 000 0, from 8517 14 000 0)
100900000695395	Tablet computers (EAEU HS from 8471 41 000 0)
100900000695396	Laptops (EAEU HS from 8471 30 000 0)
100900000695397	Printers (EAEU HS from 8443 32 100 9)
100900000695398	Liquids for electronic smoking systems (EAEU HS from 2404, from 3824 99)
100900000695399	Antifreezes (EAEU HS from 3820 00 000 0)
100900000695400	Detergents (EAEU HS from 3402)
100900000695401	Motor oil (EAEU HS from 2710 19, from 3403)
100900000695402	Low alcohol drinks (EAEU HS from 2204-2206 00, from 2208)
100900000695403	Beer (EAEU HS from 2203 00)
100900000695404	Instant coffee; instant tea (EAEU HS from 2101)
100900000695405	Vegetable oil (EAEU HS from 1507-1517)
100900000695406	Mate, or Paraguayan tea (EAEU HS from 0903 00 000 0)
100900000695407	Tea (EAEU HS from 0902)
100900000695408	Coffee (EAEU HS from 0901)
100900000695409	Juices (EAEU HS from 2009)

Note: Identifiers (codes) of goods subject to UCM marking are different in the industrial and test areas. The table shows the codes of goods subject to UMC marking used in the industrial zone.

4.1.5 "Parameter" object

Class name: Param
Parameters:

Table 4.1.5 - Structure of class metadata `Param`

Key	Data type	Description
`code`	Integer	Parameter code (identifier)
`name`	String	Name of the parameter
`type`	String	Element Type: `text` – text field; `select` - drop-down list; `directory` - directory; `checkbox` - logical, `True` or `False`, `date` – date in the `YYYY-MM-DD` format; `dynamic` - parameter with a dynamic default value
`value`	String, Integer, Float, Date, Boolean, Object	The value of the parameter depends on its type
`list`	Array	List of allowable values
`display_value`	String	Displayed value, presentation
`units`	Object	Unit of measurement
`is_required`	Boolean	Indicator of mandatory filling: `True` - prohibition of blank values; `False` - value can be blank.
`is_editable`	Boolean	Indicator of the possibility of values editing by method `item/edit`: `True` - allowed; `False` - forbidden.
`is_show_catalog`	Boolean	Indicator of presentation (output, display) of the parameter in the information on goods in the catalog: `True` - displayed; `False` – is not displayed.

4.1.6 "Order" object

Class name: `Order`
Parameters:

Table 4.1.6 - Metadata structure of the `Order` class

Key	Data type	Description
`id`	String, Integer	Order number (identifier)
`created_at`	Date	Date and time of order creation
`completed_at`	Date	Date and time of order execution
`updated_at`	Date	Date of the last update of the information
`count`	Integer	Total number of codes ordered
`comment`	String, Object	Comment to the order
`parent_order`	Integer	Number (ID) of the order based on which the current order was created
`gtin`	String	Product GTIN
`status`	Object	The current status of the order. Class object `OrderStatus` (see cl. 4.2.4)
`type`	Object	Type of ordered codes. Class `LabelType`

Key	Data type	Description
		object (see cl. 4.2.2)
`file`	Object	File for downloading codes. Class "File" object (see cl. 4.1.14)
`user`	Object	The user who created the order. Class `User` object (see cl. 4.1.1)
`item`	Object	Description of the product specified by GTIN. Class `Item` object (see cl. 4.1.4)
`labels`	Array of String	Array of ordered codes

4.1.7 "Label" object

Class name: `Label`
Parameters:

Table 4.1.7 - Metadata structure of the `Label` class

Key	Data type	Description
`uuid`	String	Identifier of the code or character
`snomer`	String	Serial number included in the code or series and number of the form (character)
`is_my`	Boolean	A sign of code ownership: `True` – to the current participant; `False` – to another counterparty
`is_shipped`	Boolean	A sign of being on the way: `True` – there has been a change of ownership, but the code has not yet been accepted; `False` - the code has been entered or it belongs to the current participant
`is_aggregate`	Boolean	Aggregation indicator (only for aggregation codes): `True` - the code is used as an aggregate (has other marking codes or low-level aggregation codes "linked" to it); `False` - the code is not a part of the aggregation ("empty")
`agg_count`	Integer	Number of codes that are currently aggregated by this code. When the code is cancelled, the number goes to `0`.
`type`	Object	Type of code. Class `LabelType` object (see cl. 4.2.2)
`status`	Object	Current code status. Class `LabelStatus` object (see cl. 4.2.3)

4.1.8 "Shipment" object

Class name: `Ship`
Parameters:

Table 4.1.8 - Structure of class metadata `Ship`

Key	Data type	Description
`code`	Integer	Shipment number (identifier)
`name`	String	Accompanying document number
`created_at`	Date	Date and time of shipment creation
`comment`	String	Comment to the order
`count`	Integer	Total number of codes specified on shipment in the parameter `labels` during method activation `/ships/add`
`agg_count`	Integer	Including the number of aggregation codes
`real_count`	Integer	Total number of codes shipped (including marking codes and aggregation codes at all nesting levels)
`taked_at`	Date	Date and time of code acceptance (data presentation) by the consignee
`received_count`	Integer	Total number of codes entered (including marking codes and aggregation codes at all nesting levels)
`price`	Object	Shipment cost estimate
`price`	Float	Total cost of the goods shipped
`currency`	String	Currency of the shipment. Enumeration value code `currency` received by the method `/directories/{:code}` (see cl. 3.8.3; the list of values is given in Table 4.3.2)
`contract`	Object	Contract (agreement). Class `Contract` object. If absent - `False`
`items`	Array of Object	Array containing descriptions of the shipped goods. Class `Item` objects (see cl. 4.1.4)
`labels`	Array	Array of information on the types of codes shipped and their quantity
`id`	Integer	Code type identifier
`name`	String	Code type name
`count`	Integer	Number of the specified type codes shipped
`operation_date`	Date	Date of shipment (as a rule, the date of the accompanying document).
`document_id`	String	UID, a unique identifier of a shipment document
`shipping_doc`	Object	Accompanying document type. Enumeration value code `shippingDocs` received by the method `/directories/{:code}` (see cl. 3.8.3; the list of values is given in Table 4.3.3)
`status`	Object	Current shipment status. Class `ShipStatus` object (see cl. 4.2.6)
`user_from`	Object	Information on the counterparty (consignor). Class `Agent` object (see cl. 4.1.2)
`user_to`	Object	Information on the counterparty (consignee). Class `Agent` object (see cl.

Key	Data type	Description
		4.1.2)
country	Object	Country to which the shipment is carried out. Enumeration value `country` received by the method `/directories/{:code}` (see cl. 3.7.3; the list of values is given in Table 4.3.1)
eas_products	Object	Additional data for the supplied goods. ***Note.*** Applicable in case of shipment data exchange in the course of the transborder trade between the EAEU member states
{:gtin}	Array of Object	The array element key name corresponds to the goods GTIN, the element value includes additional shipment parameters for the specified GTIN
product_cost	String	Price per unit of goods (including VAT, if the goods are subject to VAT)
product_tax	String	Amount of VAT per item
certificate_document_data	Array of Object	Data array for the documentary recognition of goods conformity to quality and/or safety regulations
certificate_type	String	Conformity recognition document type. `certificate_types` enumeration valure code obtained by `/directories/{:code}` (see 3.8.3; the list of values is given in Table 4.3.6)
certificate_number	String	Conformity recognition document number
certificate_date	Date	Conformity recognition document date. ***Format:*** `YYYY-MM-DD`
operation	String	Type of supply registration operation (export or import)
gtin	Array of Object	Array of volume-article accounting parameters of the supplied goods in the context of shipped GTIN and quantity. ***Note.*** Used to transfer data on the supply of goods from the "Dairy products" group
{:gtin}	Array of Object	The array element key name corresponds to the GTIN value, the value of the element contains the supply parameters for the specified GTIN.
count	Integer	Quantity of the supplied goods
certificate	String	Globally Unique Identifier (GUID) of the electronic veterinary accompanying document in SIS "AITS"
name	String	Goods name
product_cost	String	Price per unit of goods (including VAT, if the goods are subject to VAT)
product_tax	String	Amount of VAT per item
certificate_document_data	Array of Object	Data array for the documentary recognition of goods conformity to quality and/or safety regulations
certificate_type	String	Conformity recognition document type.

Key	Data type	Description
		certificate_types enumeration valure code obtained by /directories/{:code} (see 3.8.3; the list of values is given in Table 4.3.6)
certificate_number	String	Conformity recognition document number
certificate_date	Date	Conformity recognition document date. ***Format:*** YYYY-MM-DD
accepted	Object	Array of volume-article accounting parameters of the supplied goods in the context of accepted GTIN and quantity. ***Note.*** Used to transfer data on the supply of goods from the "Dairy products" group
{:gtin}	String	The array element key name corresponds to the GTIN value, the value of the element contains the supply parameters for the specified GTIN.
count	Integer	Quantity of the accepted goods
certificate	String	Globally Unique Identifier (GUID) of the electronic veterinary accompanying document
name	String	Goods name
shipment_status	Object	The current status of the dairy products supply document. ***Note:*** Displayed and updated as information about the processing of the supply document becomes available in the information system of the EAEU member state
code	String	Status symbol
message	String	Status description
debug_message	String	Error description

4.1.9 "Guide (list, enumeration)" object

Class name: Directory
Parameters:

Table 4.1.9 - Structure of class metadata Directory

Key	Data type	Description
code	String	The key (identifier) of the directory (list, enumeration)
name	String	Name of the directory (list, enumeration)
comment	String t	Description of the directory (list, enumeration)
inner	Array of Object	Array of directory elements (list, enumeration)
code	String	The code (identifier) of the element
name	String	Element value
comment	String	Note, comment

4.1.10 "Event of the revision history" object

Class name: `Event`
Parameters:

Table 4.1.10 - Structure of class metadata `Event`

Key	Data type	Description
`date`	Date	Date and time of transaction completion
`status`	Object	Assigned code status. Class `LabelStatus` object (see cl. 4.2.3)
`action`	Object	nformation on the order, shipment or other operation which resulted in a status change of the identification code. Class `Order`, `Ship` object or the results of other operations
Only for aggregation codes and aggregation operations		
`total_count`	Integer	Total number of codes listed in the aggregation
`add_count`	Integer	Number of codes added to the aggregation
`delete_count`	Integer	Number of codes excluded from the aggregation

4.1.11 "Inventory sheet" object

Class name: `Inventory`
Parameters:

Table 4.1.11 – Structure of class metadata `Inventory`

Key	Data type	Description
`group`	String	Commodity group (the list of values is given in Table 4.2.1.2)
`name`	String	Inventory sheet number
`date`	Date	Inventory sheet date
`quantity`	Integer	Total leftovers according to an inventory sheet
`Id`	String	Inventory sheet ID

4.1.12 "Supply" object

Class name: `Import`
Parameters:

Table 4.1.12 - Structure of class metadata `Import`

Key	Data type	Description
`document_id`	String	UID, unique identifier of delivery
`exporter_unp`	String	TIN of the organization-seller (supplier) from the Russian Federation
`exporter_name`	String	Name of the organization-seller (supplier)
`group`	String	Commodity group (the list of values is given in table 4.2.1.2)
`type`	String	Delivery type:

Key	Data type	Description
		`export` - transferred by the operator of the Russian Federation through the register of deliveries (temporary scheme); `manual` - entered by the participant of the turnover independently (confidential acceptance); `ship` - transferred by the exporter from through the cross-border shipment document
`created_at`	Date	Date and time of delivery registration in the SIS "Electronic Mark"
`user_to`	Object	Information on the counterparty (consignee). Class `Agent` object (see cl. 4.1.2)
`document_date`	Date	Date of the document confirming the purchase of the goods
`document_number`	String	Number of the document confirming the purchase of the goods
`sender_country_code`	String	The numeric code of the country where the product was purchased. Code of the `country` enumeration value obtained by the `/directories/{:code}` method (see clause 3.8.3; the list of values is given in Table 4.3.1)
`info`	Object	Delivery processing results
`received`	Integer	Total number of accepted identification codes
`not_received`	Integer	Total number of rejected identification codes
`status`	Object	The current processing status of the shipment. Object of the `ImportStatus` class (see section 4.2.9)
`code`	String	Status code
`message`	String	Status name
`labels`	Array of String or Array of Objects	Array of identification codes received from a supplier from the Russian Federation
`label`	String	Identification code
`status`	Integet	The result of processing the identification code: `0` - rejected; `1` - accepted

4.1.13 "Request for the manufacture of packaging" object

Class name: `TypographyOrder`
Parameters:

Table 4.1.13 - Structure of class metadata `TypographyOrder`

Key	Data type	Description
`id`	Integer	Request number
`created_at`	Date	Date and time of request creation
`group`	String	Commodity group (the list of values is given in Table 4.2.1.2)

Key	Data type	Description
quantity	Integer	Total number of packages ordered
document_id	String	UID, unique identifier of the request
comment	String	Comment to the order
typography	Object	Printing house. Object of the Agent class (see section 4.1.2)
item	Object	Description of goods. Object of the Item class (see section 4.1.4)
status	Object	The current status of the request. Object of the RquestStatus class (see section 4.2.10)
user	Object	The user who created the request. Object of the User class (see section 4.1.1)

4.1.14 "File" object

Class name: File
Parameters:

Table 4.1.14 - Structure of class metadata File

Key	Data type	Description
filename	String	The full name of the file with the URL of the download link
downloads	Integer	Number of file downloads

4.2 System directories

4.2.1 "Product Groups" directory

Class name: Group
Parameters:

Table 4.2.1 - Structure of class metadata Group

Key	Data type	Description
code	String	Key (identifier) of the product group
name	String	Name of the product group
articul_required	Boolean	Indication of the need to fill in the "Article" field for products of this group: False – not required; True - required
gtin_required	Boolean	Indication of the need to fill in the "GTIN" field for products of this group: False – not required; True - required
is_manually_add	Boolean	Indicates whether new products of this group can be added to the catalog, the /items/add method is used: False - forbidden; True - allowed
is_gtin_add	Boolean	Indicates whether new products of this group can be added to the catalog, the

Key	Data type	Description
		`/items/addByGtin` method is used: `False` - forbidden; `True` - allowed
`is_show_gtin`	Boolean	Indication of the display (output, display) of GTIN in the product information in the catalog: True - displayed; False – not displayed.
`is_show_articul`	Boolean	Indication of the display (output, display) of the article in the product information in the catalog: True - displayed; False – not displayed.

The values of the "Product Groups" directory are listed in Table 4.2.1.2.

Table 4.2.1.2 - List of product groups

`code` / Code	`name` / Name
`shoes`	`Shoes`
`milk`	`Dairy products`
`tires`	`Tires and treads`
`perfum`	`Perfumery`
`clothes`	`Consumer industry`
`photo`	`Cameras and flash lamps`
`ukz`	`Goods subject to UCM marking`
`water`	`Packaged water`
`leftovers_shoes`	`Leftovers. Footwear`
`leftovers_tires`	`Leftovers. Tyres and wheel coverings`
`leftovers_clothes`	`Leftovers. Consumer goods`

4.2.2 "Types of codest, characters" directory

Class name: `LabelType`
Parameters:

Table 4.2.2.1 - Metadata structure of the `LabelType` class

Key	Data type	Description
`id`	Integer	Type identifier of code or character
`name`	String	Type name of code or character
`method`	String	Method of generating the specified type of code
`is_bso`	Boolean	A sign of a blank document or a document with a certain degree of protection (secured document), including one containing an applied means of identification: `False` – is not a protected document; `True` – is a protected document
`digits_count`	Integer	The length of the code or the number of digits of the document number with a certain

Key	Data type	Description
		degree of protection of the specified type
letters_count	Integer	The number of characters in the document series with a defijned degree of protection
is_autotake	Boolean	Indicates whether the code can be automatically entered when transferred to another owner during shipment or sale: False – not applicable; True - applicable

The values of the "Types of codes, characters" directory are listed in Table 4.2.2.2.

Table 4.2.2.2 - List of types of codes, characters

id / Code	name / Name
1	Protection character (17 x 18)
2	Aggregation Code (DataMatrix)
3	Protected physical data carrier (34 x 17)
4	Protected physical data carrier (147 x 99)
5	Serial number
6	Aggregation Code (SSCC)
7	The code of the EAEU countries for product identification
8	Protection character containing identification code (17 x 18)
9	Protected product label (64 x 53)
10	RB code of Identification of jewellery
11	Sticker containing identification code (17*18)
12	RB code of identification of dairy products
13	Aggregation code (import)
14	Serial number for dairy products
15	Serial number (Photo products)
16	The code of the EAEU countries for the identification of goods (photographic goods)
17	Unified control mark (18*18)
18	Sticker containing identification means (18*18)
19	Protection character (18 x 10)
20	RB code of product identification (footwear, tyres and wheel coverings, consumer goods)
21	Protected physical medium of identification means, 28×30 mm

4.2.3 "Status codes" directory

Class name: LabelStatus
Parameters:

Table 4.2.3.1 - Metadata structure of the LabelStatus class

Key	Data type	Description
code	Integer	Code status code (ID)
name	String	Name of the status code

Values of the "Code statuses" directory are listed in Table 4.2.3.2.

Table 4.2.3.2 - List of code statuses

code / Code	name / Name
0	Generated
10	Ordering codes
15	Code issued
20	Manufactured
30	Validated
40	Signed by RF
45	Imported
50	The product is marked
52	Transit goods
55	Introduced into turnover in Belarus
60	Arrived from abroad
80	Sold in retail network
90	Left abroad

4.2.4 "Order statuses" directory

Class name: OrderStaus
Parameters:

Table 4.2.4.1 - Metadata structure of the OrderStaus class

Key	Data type	Description
code	Integer	Order status code (ID)
name	String	Name of the order status

The values of the "Order statuses" directory are listed in Table 4.2.4.2

Table 4.2.4.2 - List of order statuses

code / Code	name / Name
0	Created
10	Accepted from user, queued
15	Sent to RF for signature, awaiting
20	RF issued codes
30	Completed
40	Cancelled

4.2.5 "Statuses of the reports on marking" directory

Class name: ReportStaus
Parameters:

Table 4.2.5.1 - Metadata structure of the ReportStaus class

Key	Data type	Description
code	Integer	Report status code (ID)
name	String	Name of the report status

Values of the "Statuses of the reports on marking" directory are listed in Table 4.2.5.2.

Table 4.2.5.2 - List of statuses of the reports on marking

code / Code	**name / Name**
0	Created
10	In processing
20	Confirmed
30	Completed
40	Error

4.2.6 "Shipment statuses" directory

Class name: ShipStaus
Parameters:

Table 4.2.6.1 - Structure of class metadata ShipStaus

Key	**Data type**	**Description**
code	Integer	Shipment status code (identifier)
name	String	Shipment status name

Values of the "Shipment statuses" directory are listed in Table 4.2.6.2.

Table 4.2.6.2 - List of shipment statuses

code / Code	**name / Name**
0	Created
10	On the way
12	Shipped
15	Error
17	Acceptance pending
19	Partially entered
20	Accepted
30	Cancelled

4.2.7 "Participants roles" directory

Class name: UsersRoles
Parameters:

Table 4.2.7.1 - Structure of class metadata UsersRoles

Key	**Data type**	**Description**
code	Integer	Code (ID) of the role
description	String	Name (description) of the role

Some values of the "Participants roles" directory are listed in Table 4.2.7.2.

Table 4.2.7.2 - List of participants roles (fragment)

code / Code	**description / Description**
admin	Administrator

code / Code	description / Description
exporter	Merchandise turnover participant
head	Headquaters
ukz	Participant of UCM turnover

4.2.8 "Report types" directory

Class name: ReportType
Parameters:

Table 4.2.8.1 - Structure of class metadata ReportType

Key	Data type	Description
id	Integer	Report code (identifier)
name	String	Report name

Values of the "Report types" directory are listed in Table 4.2.8.2.

Table 4.2.8.2 - List of report types

id / Code	name / Name
1	Introduction into turnover
2	Aggregation
3	Manufacture
4	Use of CPD (cryptography device)
5	Manufacturing for UOT
6	Introduction into turnover. Leftovers
7	Introduction into turnover in Belarus
8	Cancel application
10	Labeling of goods made from customer's raw materials
13	Changing the marking parameters

4.2.9 "Supply statuses" directory

Class name: ImportStatus
Parameters:

Table 4.2.9.1 - Structure of class metadata ImportStatus

Key	Data type	Description
code	Integer	Supply status code (identifier)
name	String	Supply status name

Values of the "Supply statuses" directory are listed in Table 4.2.9.2.

Table 4.2.9.2 - List of supply statuses

code / Code	name / Name
0	Created
5	In processing
10	Processed
20	Posted
30	Credited

code / Code	name / Name
40	Error

4.2.10 "Request statuses" directory

Class name: RequestStatus
Parameters:

Table 4.2.10.1 - Structure of class metadata RequestStatus

Key	Data type	Description
code	Integer	Request status code (identifier)
name	String	Request status name

Values of the "Request statuses" directory are listed in Table 4.2.10.2.

Table 4.2.10.2 - List of request statuses

code / Code	name / Name
20	Confirmed
40	Error

4.3 Enumerations

4.3.1 "Countries" enumeration

List code: country

Some values of the "Country" enumeration listed in Table 4.3.1.

Table 4.3.1 - List of countries (fragment, EAEU countries)

code / Code	name / Name
051	Armenia
112	Belarus
398	Kazakhstan
417	Kyrgyzstan
643	Russia

4.3.2 "Currencies" enumeration

List code: currency

Some values of the "Currencies" enumeration are listed in Table 4.3.2.

Table 4.3.2 - Enumeration of currencies (fragment)

code / Code	name / Name
byn933	Belarusian Ruble
eur978	Euro
kzt398	Kazakh Tenge
rub643	Russian Ruble

code / Code	name / Name
usd840	US Dollar

4.3.3 "Document types" enumeration

List code: shippingDocs

Some values of the "Document types" enumeration are listed in Table 4.3.3.

Table 4.3.3 - List of document types (fragment)

code / Code	name / Name
akt	Write-off act
aktnedost	Deficiency act
aktreorg	Certificate of transfer
invoice	Invoice, preliminary invoice, commercial invoice
customs	Customs permit
statdeclaracia	Intrastat
ttn	BL – bill of lading
tnttn	GCN – goods consignment note
ettn	EBL – electronic bill of lading
etn	EGCN – electronic goods consignment note

4.3.4 "Causes of cancellation" enumeration

List code: reasons

Some values of the "Causes of cancellation" enumeration are listed in Table 4.3.4.

Table 4.3.4 - List of causes of cancellation (fragment)

code / Code	name / Name
damage	Spoiling
destruction	Destruction
liquidation	Liquidation (termination of activity)
lost	Loss
mns	Regulator Notice (Ministry of Taxes and Duties)
other	For production needs
spoilage	Production fault
theft	Stealing
usemark	Application (labelling)

4.3.5 "Reasons for introduction into turnover" enumeration

List code: circulation_reason

Values of the "Reasons for introduction into turnover" enumeration are listed in Table 4.3.5.

Table 4.3.5 – List of reasons of introduction into turnover

code / Code	**name / Name**
import	Import from EAEU member states
reversion	Return to retail

4.3.6 "Certificate types" enumeration

List code: certificate_types

Values of the "Certificate types" enumeration are listed in Table 4.3.6.

Table 4.3.6 – List of conformity recognition documents (certificates)

code / Code	**name / Name**
CONFORMITY_CERTIFICATE	Certificate of conformity
CONFORMITY_DECLARATION	Declaration of conformity
STATE_REGISTRATION_CERTIFICATE	State registration certificate

5 ERROR CODES AND THEIR DESCRIPTION

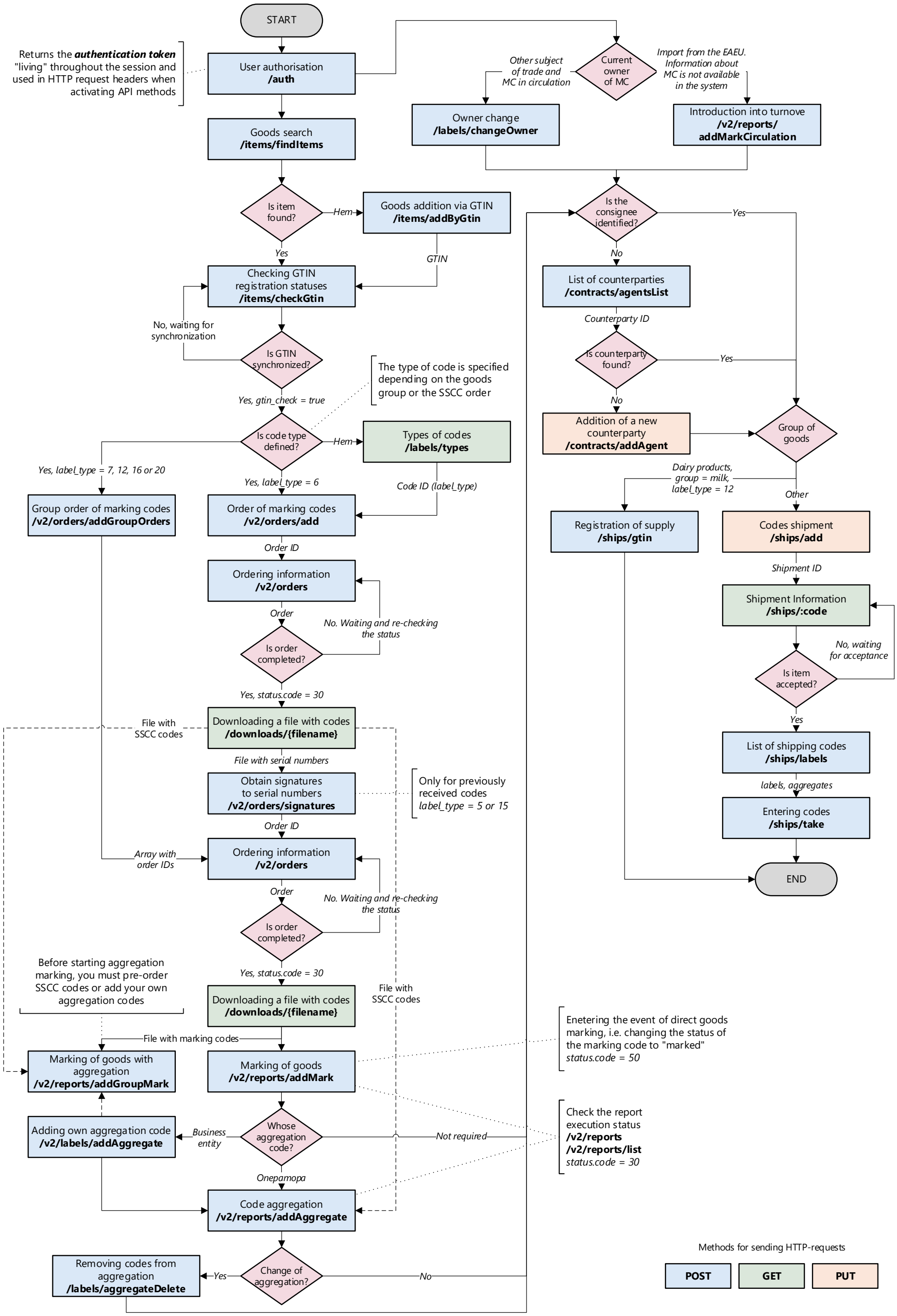
The HTTP response may return error codes with corresponding descriptions listed in Table 5.1.

Table 5.1 - Error codes and their description

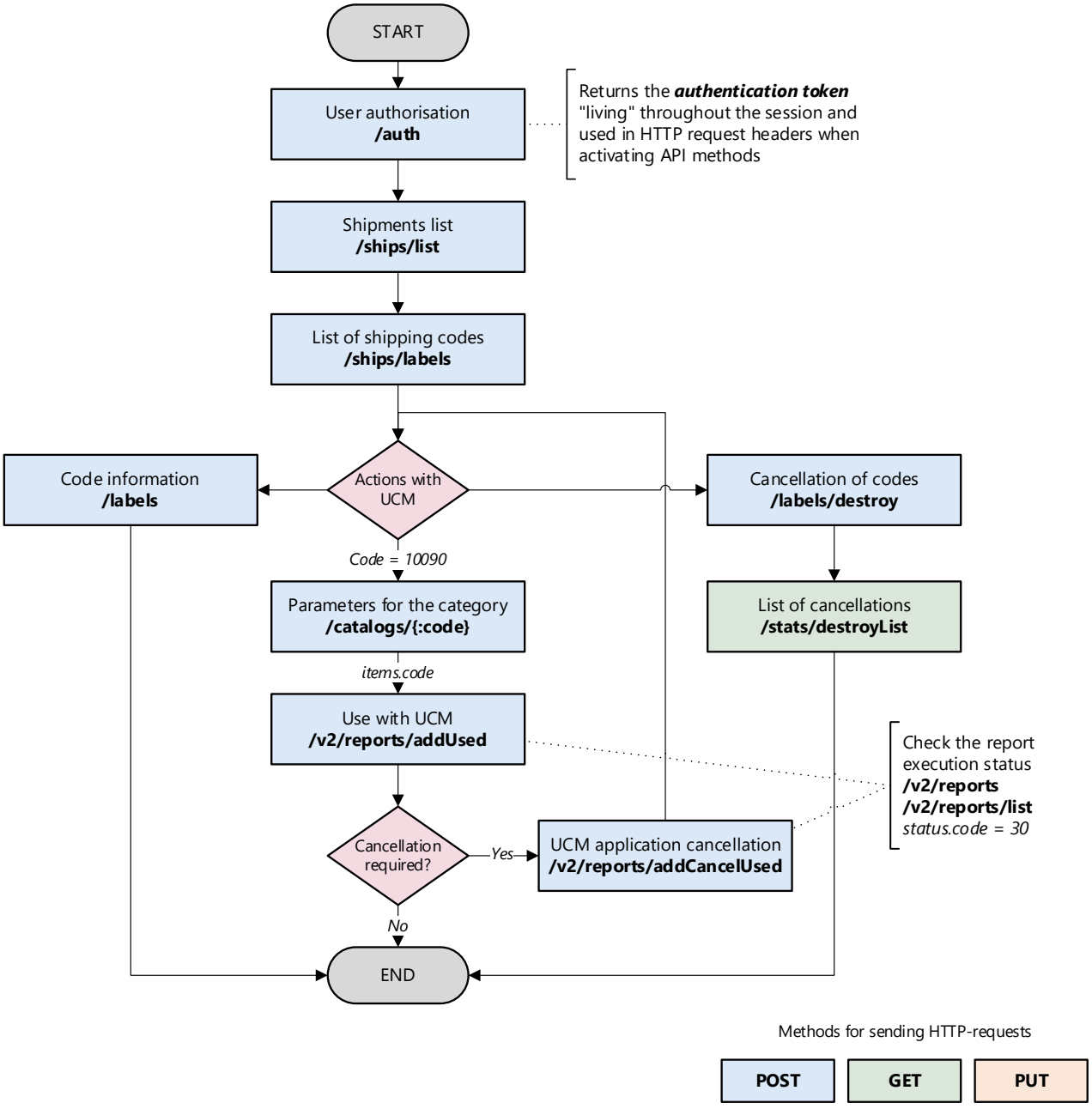
Error code	**Description**
HTTP 400 response code. Incorrect data	
400	Error in request input parameters
HTTP 401 response code. Invalid authorization token	
1202	Incorrect user name
1203	Incorrect password
10004	Authorization token has expired
13001	Invalid token
13002	Invalid token or the user is blocked
HTTP 403 response code. No access rights	
403	The file was not found or you do not have access rights
13003	Access Denied / Access forbidden
2102	You cannot create this order
HTTP 404 response code. The requested data was not found	
1250	Directory enumeration not found
1451	Counterparty not found
1452	Cannot add a counterparty
1453	Contract (agreement) not found
1454	Cannot update the data of the contract (agreement)
1455	A counterparty with such data is already present in the system
1450	Cannot update the status codes
1488	Shipment not found
1501	Order not found
1507	Cannot save the codes from the order

Error code	Description
1508	Product groups error
1509	The codes must belong to you. The status of the code should be 'Code issued'. GTIN must correspond to the product group
1660	The product catalog section was not found
1701	Labelling code or aggregation code not found
1702	Error creating/updating code information
1706	The specified code is not an aggregation code
1708	Invalid code type
1711	Aggregation code cannot be added
1766	Report not found
2001	Product not found
2004	Code type not found
2005	Error updating product information
2007	Incorrect product parameters
2008	Cannot save the product parameters
2010	Product GTIN not found
2011	EAEU HS Code not specified
2012	Mandatory product specifications not specified
10001	Delivery document not found or previously processed

APPENDIX A
Diagram of the API method activation sequence when marking goods by means of identification



APPENDIX B
Diagram of the API method activation sequence when marking goods with unified control marks



APPENDIX C
Description of business processes of marking in separate commodity groups

C.1 Common features of goods marking processes by means of identification

For the goods not yet subject to mandatory labelling with ID marks in Belarus, or those not subject to labelling in Belarus but subject thereto in other EAEU member states (perfumes, photography accessories, packaged water), basic API methods are applicable generally, which methods are initiated in the following sequence:

C.1.1 A group order for labelling codes in respect of all products subject thereto, in the adequate quantity for each commodity item, is created by `/v2/orders/addGroupOrders` method with code type `label_type = 7` (for all commodity groups) or `label_type = 16` (for photography accessories) and with respective `"group"` parameter value. Once the order is successfully fulfilled, the labelling codes are assigned with “Goods labelled” status (`status = 50`).

C.1.2 Should `/v2/orders/add` method with code type `label_type = 5` (for all commodity groups) or `label_type = 15` (for photography accessories) succeed to generate serial numbers, then signatures to the obtained serial numbers shall be requested for by `/v2/orders/signatures` method with respective `"group"` parameter value, and an order to generate labelling codes for each requested GTIN shall be produced.

Note: Serial numbers generated before 01.11.2021, which are incompatible with the adopted format, i.e. include no issuing EAEU member state numerical ID in the first digit, are invalid and may not be used to order labelling codes by `/v2/orders/signatures` method.

C.1.3 To order labelling codes, `/v2/orders/addGroupOrders` method shall be used with the code types specified in C.1.1. The methods named in C.1.2 to order serial numbers and to request signatures thereto remain in the current API version for compatibility purposes, but will be excluded in the higher versions.

C.1.4 `/v2/orders/statusList` method is used to review the current status of the created orders.

C.1.5 `/v2/orders` method is used to return an array of the generated labelling codes by the order ID, if the order status is `30`.

C.1.6 A labelling report (i.e. a report of labelling code application through conversion into GS1 Datamatrix 2D barcodes, manufacturing and actual attachment of the ID marks to the goods) is not required to be submitted.

C.1.7 `/labels/changeOwner` method (for all commodity groups except packaged water) may be used to change the current labelling code owner, if required, before the labelled goods are shipped, provided that such labelling codes are not allocated to any turnover agent (business) and the data on such labelling codes is available in SIS “Electronic Mark”.

C.1.8 The data on labelled goods shipment (transfer, transporting) by the consignor to a counterparty (business), including export transactions, are exchanged by `/ships/add` method.

C.1.9 The consignee accepts the labelled goods in full or in part by `/ships/take` method.

C.2 Labelling particularities for dairy products, footwear products, tyres, consumer goods

The following API methods shall be used for commodity groups “Dairy products” (`group = milk`), “Footwear products” (`group = shoes`), “Tyres and wheel coverings” (`group = tires`), “Consumer goods” (`group = clothes`). The methods are initiated in a certain sequence and with consideration of the particularities described below:

C.2.1 A group order for labelling codes in respect of all products subject thereto, in the adequate quantity for each commodity item, is created by `/v2/orders/addGroupOrders` method with code type `label_type = 12` (for dairy products) or `label_type = 20` (for footwear, tyres and wheel coverings) Once the order is successfully fulfilled, the labelling codes are assigned with “Validated” status (`status = 30`).

C.2.2 `/v2/labels/addAggregate` method is used to add a list of own aggregation codes created by a turnover agent.

C.2.3 A labelling report is sent by `/v2/reports/addMark` method or a group labelling report is sent by `/v2/reports/addGroupMark` method. In the latter case the ID mark application reporting is accompanied with primary packaging aggregation, i.e. product ID label application and packaging aggregation code application are reported, while the product ID codes and the packaging aggregation codes are getting interlinked (in other words, this method combines `/v2/reports/addMark` method with `/v2/reports/addAggregate` method). Report processing involves control of `label_type = 20` code, labelling code status “Validated” (`status = 30`), and labelling code assignment to any goods turnover agent (i.e. the business that requests and receives the labelling code).

C.2.4 While submitting a labelling report or a group labelling report, an array of labelling parameters (`"params"`), the list and IDs (codes) of which shall be specified, as well as the types of their values, may be obtained from `"mark_params"` attribute returned in the response of `/catalogs/{:code}` method, where `code = 10002` (“Footwear” group), `code = 10089` (“Dairy products” group), `code = 10086` (“Tyres and wheel coverings” group), `code = 10081` (“Consumer goods” group), i.e. from the request for information about the respective goods catalogue section (category) and the list of its parameteres.

C.2.5 The labelling parameters (information about the labelled goods) recorded in a labelling report are listed in Appendix D.

C.2.6 Report status may be checked by `/v2/reports` method or `/v2/reports/list` method using the report UID. Once successfully reported, the labelling codes are assigned with “Goods labelled” status (`status = 50`).

C.2.7 `/ships/gtin` method is used to submit data on dairy products shipment/supply to Russia by GTIN and quantity (registering by volume, article numbers, grades). `/ships/gtinList` method returns the list of effected supplies.

C.3 Features of goods marking processes with unified control marks

Both general and special API methods are used for products subject to UCM marking (`group = ukz`), activated in the following sequence:

C.3.1 Methods `/ships/list` and `/ships/labels` obtain a list of the series and numbers of the UCM forms provided to the turnover participant.

C.3.2 Method `/catalogs/{:code}` with `code = 10090` which corresponds to the request for information in relation to the catalog section (category) of goods marked with the UCM returns a list of relevant goods and their codes in the `items` array.

C.3.3 Method `/v2/reports/Added` transfer a report on the actual use of the UCM (application to the goods).

C.3.4. Method `/v2/reports/addCancelUsed` can be used to completely or partially cancel UCM application, where required.

C.3.5 The UCM status can be checked via the UCM usage report UID through the `/v2/reports` or `/v2/reports/list` methods.

C.4 Labelling particularities for goods leftovers

Goods leftovers are labelled by the following API methods, initiated in a certain sequence and with consideration of the particularities described below:

C.4.1 Goods leftovers inventory results are submitted by `/v2/items/inventory` method with an appropriate value of `"group"` parameter.

C.4.2 Labelling codes are ordered in required quantity by `/v2/orders/addGroupOrders` method with `label_type = 20` code. Provided that, either GTIN for goods leftovers (Table B.4.1) without their full description (applicable to the goods leftovers intended to be distributed exclusively in Belarus without exporting), or GTIN for respective article numbers with their full description, is used. the labelling codes are assigned with “Validated” status (`status = 30`).

Table C.4.1 – GTIN used to label goods leftovers

GTIN	Description
04814130000030	Leftovers. Footwear products. ONLY for distribution in Belarus
04814130000047	Leftovers. Tyres and wheel coverings. ONLY for distribution in Belarus
04814130000054	Leftovers. Consumer goods. ONLY for distribution in Belarus

C.4.3 The current status of created orders is reviewed by `/v2/orders/statusList` method.

C.4.4 `/v2/orders` method is used to return an array of the generated labelling codes by the order ID, if the order status is `30`.

C.4.5 Leftover goods labelling report is submitted by`/v2/orders/addMarkLeftovers` method. While submitting reports, labelling codes with code type `label_type = 7` or `label_type 20` may be used. Provided

that, for a labelling code with `label_type = 20` code type, the assignment to a turnover agent (i.e. a business which orders and obtains the labelling code) should be additionally checked.

C.4.6 Report status may be checked by `/v2/reports` method or `/v2/reports/list` method using the report UID. Once successfully reported, the labelling codes are assigned with “Introduced into turnover in Belarus” status (`status = 55`).

C.5 Particularities of acceptance and introduction into turnover of goods procured in the course of transborder trade

For the goods marked with ID labels, procured in other EAEU member states (in particular, in the Russian Federation) as part of cross-border trade, acceptance and putting into turnover operations are carried out taking into account the following features:

C.5.1 Using the /`ships/importList` and /`ships/import` methods, the importer (consignee) checks the availability and obtains information about the shipment (supply) of the goods to its address.

C.5.2 Depending on the method used for preparing and transmitting information about deliveries, this information can be processed in one of the following modes:

C.5.2.1 Provisional order.

C.5.2.1.1 The goods are withdrawn from turnover by the exporter (consignor) and acquire the status "Exported to the EAEU". Information about the goods sold to the importer (consignee) by identification codes is formed and transmitted through the register of deliveries by the national operator of the marking information system of the EAEU member state where the goods were purchased. After the specified information is transmitted, it becomes available in the SIS "Electronic Mark" (delivery type `type = export`).

C.5.2.1.2 The importer (consignee) enters information about the actual acceptance of the goods purchased by him one-time or in several operations, dividing the delivery into several parts (batches), using one of the following methods:

putting into circulation by marking codes using the `/v2/reports/addMarkCirculation` method;

trust acceptance by identification codes using the `/ships/addImport` method (in the future, such goods, if necessary, can be put into circulation by marking codes).

C.5.2.1.3 Regardless of the chosen method of acceptance, the `"document_id"` attribute must contain the UID of the incoming document, which has the delivery type `type = export`, according to which acceptance is carried out.

C.5.2.1.4 After processing the information confirming acceptance, in the SIS "Electronic Mark" the product is assigned the status "Imported to the Republic of Belarus", while its status does not change in the information system of marking of the EAEU member state from which the product was imported.

C.5.2.2 Cross-border shipment.

C.5.2.2.1 The exporter (consignor) in the information system for marking goods of the EAEU member state from which the goods are imported forms and sends to the importer (consignee) information about the delivery of goods (cross-border shipment document), which are automatically sent to the SIS "Electronic Mark" (delivery type `type = ship`). The goods are in the "Acceptance pending" status.

C.5.2.2.2 The importer (consignee) can immediately accept the goods through putting into turnover using marking codes or first perform trusted acceptance using identification codes and then, if necessary, put the goods into turnover using marking codes. After processing the information about the cross-border shipment by the importer (consignee) and sending the result from the SIS "Electronic Mark" to the information labeling system of the EAEU member state of the exporter (consignor), the goods receive the status "Exported to the EAEU" in case of successful acceptance or the status "In circulation" is restored in case of refusal from acceptance.

C.5.3 If the initial recognition of goods with the status "Acceptance pending" is performed on the basis of a cross-border shipment document, then:

when using the report submission method `/v2/reports/addMarkCirculation` or the trusted acceptance method `/ships/addImport`, the `"document_id"` attribute must contain the UID of the document with the delivery type `type = ship`, that is used to carry out acceptance;

the acceptance operation is performed once for the entire document as a whole by one batch indicating all the identification codes actually accepted from this delivery. The identification codes that are not listed in the trusted acceptance or codes that have no marking codes corresponding to them in the putting into turnover report, are automatically rejected and returned to the supplier (i.e. the product is not withdrawn from circulation in the territory of the Russian Federation and the status “In circulation” is restored to it);

when trying to retransmit information about the acceptance of goods using the same UID of the delivery document (`type = ship`), the SIS "Electronic sign" will generate an error about the absence of a delivery document awaiting acceptance (error code 10001).

C.5.4 Using the `/v2/reports/addMarkCirculation` method, indicating the reason “Import from the EAEU countries” (`circulation_reason = import`), the goods with the means of identification that arrived (imported) into the territory of the Republic of Belarus are put into turnover, while:

in the "`params`" section of the report on putting into turnover are indicated: the numeric code of the country from where the goods are imported from; date and number of the document confirming the purchase of the goods; TIN and name of the supplier (seller organization);

since the deliveries from the Russian Federation have identification codes without a cryptopart (the so-called “cryptotail”), the list of accepted marking codes in

the full format must be indicated in the "`labels`" attribute of the report on putting into turnover;

it is allowed to put the goods into turnover by several operations (except for the cases described in clause C.5.3), dividing the delivery into several parts (batches), if trusted acceptance was done for the entire cross-border shipment or the goods have the status "Exported to the EAEU".

C.5.5. Using the `/v2/reports` or `/v2/reports/list` methods, you can check the status of the putting into turnover report by UID. Upon successful completion of the operation of putting into turnover (receipt) of the goods, the marking codes are assigned the status “Introduced into turnover in the Republic of Belarus” (`status = 55`).

C.6 Labelling particularities for goods made from customer's raw materials

For the goods made from customer's raw materials, labeling operations are performed taking into account the following features:

C.6.1 Labeling codes are generated by the customer (toller) in the marking information system of the EAEU member state in which he is registered (for example, in the Russian Federation) and transferred to the contractor (processor, resident of the Republic of Belarus) within the framework of the tolling agreement concluded between him. At the same time, the services of the SIS "Electronic Mark" are not used.

C.6.2 Using the /v2/reports/addMarkTrans method, the processor sends a report on the marking of goods produced from customer's raw materials.

C.6.3 Report status may be checked by `/v2/reports` method or `/v2/reports/list` method using the report UID. Once successfully reported, the labelling codes are assigned with “Transit goods” status (`status = 52`).

C.6.4 Goods with the current status "Transit goods":

is not subject to sale on the territory of the Republic of Belarus and is acceptable for circulation only in terms of transportation and storage;

in the future, such a goods can be put into turnover as a goods acquired in the framework of cross-border trade.

APPENDIX D
Labelling report parameters for different commodity groups

Table D.1 – Information about goods recorded in a labelling report

Labelling parameter[1]		**Commodity group (Appl.**[2]**/Resp.**[3]**)**			
code / Code	**name / Name**	**Dairy products**	**Footwear products**	**Tyres and wheel coverings**	**Consumer goods**
1. "Production" mode					
25	Date of manufacture/production	+ / Yes	+ / No	+ / No	+ / No
90	Batch number	+ / No	+ / No	+ / No	+ / No
91	Variable weight (value, unit of measurement)	+ / No	–	–	–
92	Expiration date	+ / Yes	–	–	–
93	Production order type	+ / Yes	+ / Yes	+ / Yes	+ / Yes
94	Goods name	+ / No	+ / No	+ / No	+ / No
95	Marking method (print protection)	+ / Yes	+ / Yes	+ / Yes	+ / Yes
101	Purpose of labelling	–	+ / Yes, with value "1"	+ / Yes, with value "1"	+ / Yes, with value "1"
108	Purpose of dairy products marking	+ / Yes, with value "1"	–	–	–
131	VSD number	+ / No	–	–	–
132	UI VSD	+ / No	–	–	–
2. "Import from other countries" mode					
25	Date of manufacture/production	+ / Yes	+ / No	+ / No	+ / No
90	Batch number	+ / No	+ / No	+ / No	+ / No
91	Variable weight (value, unit of measurement)	+ / No	–	–	–
92	Expiration date	+ / Yes	–	–	–
94	Goods name	+ / No	+ / No	+ / No	+ / No
95	Marking method (print protection)	+ / Yes	+ / Yes	+ / Yes	+ / Yes
96	Goods declaration registration number[4]	+ / Yes	+ / Yes	+ / Yes	+ / Yes
97	Date of manufacture according to	+ / Yes	+ / Yes	+ / Yes	+ / Yes

Labelling parameter[1]		**Commodity group (Appl.[2]/Resp.[3])**			
code / Code	**name / Name**	**Dairy products**	**Footwear products**	**Tyres and wheel coverings**	**Consumer goods**
	the declaration				
98	Contract number	+ / No	+ / No	+ / No	+ / No
99	Contract date	+ / No	+ / No	+ / No	+ / No
100	Exporting country	+ / Yes	+ / Yes	+ / Yes	+ / Yes
101	Purpose of labelling	–	+ / Yes, with value "2"	+ / Yes, with value "2"	+ / Yes, with value "2"
108	Purpose of dairy products marking	+ / Yes, with value "2"	–	–	–
131	VSD number	+ / No	–	–	–
132	UI VSD	+ / No	–	–	–
134	Item number in the goods declaration	+ / No	+ / No	+ / No	+ / No
3. "Commission trade" mode					
25	Date of manufacture/production	–	+ / No	+ / No	+ / No
90	Batch number	–	+ / No	+ / No	+ / No
94	Goods name	–	+ / No	+ / No	+ / No
95	Marking method (print protection)	–	+ / Yes	+ / Yes	+ / Yes
101	Purpose of labelling	–	+ / Yes, with value "3"	+ / Yes, with value "3"	+ / Yes, with value "3"
102	Reference: document date	–	+ / Yes	+ / Yes	+ / Yes
103	Reference: document number	–	+ / Yes	+ / Yes	+ / Yes
104	Reason for labelling	–	+ / Yes	+ / Yes	+ / Yes
4. "Relabelling" mode					
25	Date of manufacture/production	+ / Yes	+ / No	+ / No	+ / No
90	Batch number	+ / No	+ / No	+ / No	+ / No
91	Variable weight (value, unit of measurement)	+ / No	–	–	–
92	Expiration date	+ / Yes	–	–	–
94	Goods name	+ / No	+ / No	+ / No	+ / No
95	Marking method (print protection)	+ / Yes	+ / Yes	+ / Yes	+ / Yes
101	Purpose of labelling	–	+ / Yes, with value	+ / Yes, with value	+ / Yes, with value

Labelling parameter[1]		**Commodity group (Appl.[2]/Resp.[3])**			
code / Code	**name / Name**	**Dairy products**	**Footwear products**	**Tyres and wheel coverings**	**Consumer goods**
			"4"	"4"	"4"
102	Reference: document date	+ / Yes	+ / Yes	+ / Yes	+ / Yes
103	Reference: document number	+ / Yes	+ / Yes	+ / Yes	+ / Yes
104	Reason for labelling	+ / Yes	+ / Yes	+ / Yes	+ / Yes
108	Purpose of dairy products marking	+ / Yes, with value "4"	–	–	–
124	Reason for remarking	+ / No	+ / No	+ / No	+ / No
5. "Import from the EAEU Member States" mode[5]					
25	Date of manufacture/production	+ / Yes	–	–	–
90	Batch number	+ / No	–	–	–
91	Variable weight (value, unit of measurement)	+ / No	–	–	–
92	Expiration date	+ / Yes	–	–	–
94	Goods name	+ / No	+ / No	+ / No	+ / No
95	Marking method (print protection)	+ / Yes	+ / Yes	+ / Yes	+ / Yes
101	Purpose of labeling	–	+ / Yes, with value "5"	+ / Yes, with value "5"	+ / Yes, with value "5"
108	Purpose of dairy products marking	+ / Yes, with value "5"	–	–	–
109	Date of the document confirming the purchase of the goods	–	+ / Yes	+ / Yes	+ / Yes
110	Number of the document confirming the purchase of the goods	–	+ / Yes	+ / Yes	+ / Yes
111	Taxpayer number	–	+ / Yes	+ / Yes	+ / Yes
112	Name of the seller organization	–	+ / Yes	+ / Yes	+ / Yes
126	Country of export (EAEU)	–	+ / Yes	+ / Yes	+ / Yes
131	VSD number	+ / No	–	–	–
132	UI VSD	+ / No	–	–	–

Notes:

1. Parameter IDs (codes) differ in the industrial and testing zones. The table specifies parameter codes used in the commercial zone.

2. Labelling parameter applicability to submitting a labelling report for a respective commodity group according to the goods usage mode:
"+" - applicable;
"–" - not applicable.
3. Requirement to asign a correct value to a labelling parameter while submitting a labelling report:
"Yes" - mandatory, checked;
"No" - not checked.
4. The registration number of the goods declaration must correspond to the established template:
[X...X/DDMMYY/NNNNNNN], где:
X...X – code of the customs authority that registered the declaration for goods, in accordance with the classifiers of customs authorities used in the EAEU member states (in the Republic of Armenia, the 2-digit code of the customs authority is indicated; in the Republic of Belarus, the Republic of Kazakhstan and the Kyrgyz Republic - 5–digit; in the Russian Federation - 8–digit);
DDMMYY – date of issue, signing, approval or registration of the document (day, month, last two digits of the year);
NNNNNNN – number of the customs document according to the register.
5. Applies in case of importation of goods without means of identification from a member state of the EAEU, which has not introduced mandatory labeling of such goods.

