

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 LABELING OF GOODS  
WITH UNIFIED CONTROL MARKS  
OR IDENTIFICATION MEANS  
(SIS "ELECTRONIC MARK")**

**CROSS-SYSTEM COUPLING WEB SERVICE**

**API specification**

*Labeling and circulation of goods*

Revision 4.10

Minsk  
2025

## CONTENTS

ABSTRACT .....	1
1 TERMS AND ABBREVIATIONS .....	1
2 GENERAL .....	2
2.1 Cross-system coupling interface.....	2
2.2 HTTP request format .....	3
2.3 HTTP response format.....	3
2.4 HTTP request execution procedure .....	4
2.5 Data types and formats .....	5
2.6 Escaping.....	5
3 API METHODS .....	5
3.1 User identification methods.....	5
3.1.1 Method "Authorization" .....	5
3.1.2 Method "Logout" .....	7
3.1.3 Method "Password change" .....	7
3.2 Product catalog handling methods.....	8
3.2.1 Method "Product catalog sections" .....	8
3.2.2 Method "Product category parameters" .....	9
3.2.3 Method "Item search" .....	15
3.2.4 Method "Item addition by GTIN" .....	22
3.2.5 Method "GTIN registration status check" .....	26
3.2.6 Method "Item information change" .....	27
3.2.7 Method "Inventory" .....	28
3.2.8 Method "Inventory list" .....	29
3.3 Order handling methods .....	30
3.3.1 Method "Code order" .....	30
3.3.2 Method "Order details" .....	32
3.3.3 Method "Order list" .....	36
3.3.4 Method "Group order of labeling codes" .....	41
3.3.5 Method "Order status list" .....	44
3.3.6 Method "Number of completed orders" .....	46
3.3.7 Method "Ordered code list" .....	47
3.4 Code and label handling methods.....	47
3.4.1 Method "Custom aggregation code addition" .....	47
3.4.2 Method "Deletion of codes from aggregation" .....	49
3.4.3 Method "Types of codes or labels" .....	50
3.4.4 Method "Discarding codes or labels" .....	51
3.4.5 Method "Discarding list" .....	54
3.4.6 Method "Labeling code validation" .....	56
3.4.7 Method "Information on the aggregate structure and composition" .....	58
3.5 Report handling methods .....	61

3.5.1 Method "Labeling report" .....	61
3.5.2 Method "Report on group labeling with aggregation" .....	62
3.5.3 Method "Aggregation report" .....	64
3.5.4 Method "UCM usage" .....	67
3.5.5 Method "Report execution status" .....	67
3.5.6 Method "Report list" .....	75
3.5.7 Method "Report types" .....	91
3.5.8 Method "Used code list" .....	93
3.5.9 Method "Circulation report" .....	95
3.5.10 Method "Residual stock labeling report" .....	98
3.5.11 Method "UCM cancellation" .....	99
3.5.12 Method "Report on labeling of goods made from customer-owned raw materials" .....	99
3.5.13 Method "Labeling parameter change report" .....	100
3.5.14 Method "Manufacturing parameter report" .....	101
3.5.15 Method "Owner change report" .....	104
3.5.16 Method "Report on deletion of codes from the aggregate" .....	109
3.6 Agent handling methods .....	110
3.6.1 Method "Agent details" .....	110
3.6.2 Method "Adding a new agent" .....	110
3.6.3 Method "Agent list" .....	111
3.7 Shipment handling methods .....	113
3.7.1 Method "Code shipment" .....	113
3.7.2 Method "Shipment details" .....	121
3.7.3 Method "Shipped code list" .....	123
3.7.4 Method "Code capitalization" .....	125
3.7.5 Methods "Shipment list" .....	127
3.7.6 Method "Diary product supply registration" .....	136
3.7.7 Method "Dairy product supply list" .....	142
3.7.8 Method "List of supplies from the EAEU" .....	144
3.7.9 Method "Acceptance of supply from the EAEU" .....	146
3.7.10 Method "Information about supply from the EAEU" .....	148
3.7.11 Method "Information on the supply of dairy products" .....	150
3.7.12 Method "Full acceptance of codes" .....	153
3.7.13 "List of accepted goods supplied from the EAEU" .....	154
3.8 Auxiliary methods.....	155
3.8.1 Method "File download" .....	155
3.8.2 Method "Enumeration directories list" .....	156
3.8.3 Method "Enumeration directory elements" .....	156
3.8.4 Method "Printing house list" .....	157
3.9 Billing handling methods.....	158
3.9.1 Method "Service delivery report list" .....	158
4 OBJECTS, DIRECTORIES AND ENUMERATIONS .....	161
4.1 Object classes.....	161

4.1.1 Object "User" .....	161
4.1.2 Object "Agent" .....	161
4.1.3 Object "Product catalog section (category)" .....	162
4.1.4 Object "Item" .....	163
4.1.5 Object "Parameter" .....	166
4.1.6 Object "Order" .....	166
4.1.7 Object "Shipment" .....	167
4.1.8 Object "Directory (list, enumeration)" .....	171
4.1.9 Object "Inventory" .....	172
4.1.10 Object "Import" .....	172
4.1.11 Object "Aggregate" .....	173
4.1.12 Object "Service delivery report" .....	174
4.1.13 Object "File" .....	175
4.2 System directories .....	175
4.2.1 Directory "Product groups" .....	175
4.2.2 Directory "Label types" .....	176
4.2.3 Directory "Label statuses" .....	177
4.2.4 Directory "Order statuses" .....	178
4.2.5 Directory "Report statuses" .....	178
4.2.6 Directory "Shipment statuses" .....	179
4.2.7 Directory "Participant roles" .....	179
4.2.8 Directory "Report types" .....	179
4.2.9 Directory "Import statuses" .....	180
4.2.10 Directory "Request statuses" .....	180
4.3 Enumerations .....	181
4.3.1 Enumeration "Countries" .....	181
4.3.2 Enumeration "Currencies" .....	181
4.3.3 Enumeration "Document types" .....	181
4.3.4 Enumeration "Discard reasons" .....	182
4.3.5 Enumeration "Circulation reasons" .....	182
4.3.6 Enumeration "Certificate types" .....	182
4.3.7 Enumeration "Owner change reasons" .....	183
5 ERROR CODES AND THEIR DESCRIPTIONS .....	183
ANNEX A Sequence diagram of calling API methods when labeling goods with identification means .....	185
ANNEX B Sequence diagram of calling API methods when labeling goods with unified control marks .....	186
ANNEX C Description of labeling business processes in individual product groups .....	187
C.1 General specifics of labeling of goods with identification means .....	187
C.2 Specifics of labeling processes for dairy products, footwear, tires and consumer goods .....	188
C.3 Specifics of labeling of goods with unified control marks .....	189

C.4 Specifics of residual stock labeling.....	189
C.5 Specifics of acceptance and circulation of goods purchased in cross-border trade.....	190
C.6 Specifics of labeling of goods made from customer-owned raw materials....	192
C.7 Specifics of labeling of goods that require the submission of information on manufacturing parameters.....	192
C.8 Specifics of business entity reorganization processes .....	193
ANNEX D Parameters of labeling report in individual product groups.....	195
ANNEX E Labeling code templates for individual product groups .....	200
ANNEX F Authorization documents .....	204

## **ABSTRACT**

This document describes the procedure for cross-system coupling of information systems of participants in the circulation of goods subject to labeling with identification means or unified control marks with the hardware and software complex of SIS "Electronic Mark" via API protocol.

## **1 TERMS AND ABBREVIATIONS**

The following terms and abbreviations are used in this document:

**AC** (aggregation code): a unique sequence of characters used for aggregating and forming package identification means;

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

**EAEU** (Eurasian Economic Union): international organization for regional economic integration with international legal personality established by the Treaty on the Eurasian Economic Union. Current member states include the Republic of Armenia, the Republic of Belarus, the Republic of Kazakhstan, the Kyrgyz Republic and the Russian Federation;

**EAEU CN of FEA**: EAEU Commodity Nomenclature of Foreign Economic Activity;

**GTIN** (Global Trade Item Number): international code for labeling, identification and accounting of goods;

**HTTP** (HyperText Transfer Protocol): application layer protocol for the transfer of data between the client and the server. TCP port 80 is used by default;

**HTTPS** (HyperText Transfer Protocol Secure): an extension of the HTTP protocol supporting encryption for enhanced security. Data is transmitted over TLS cryptographic protocols. TCP port 443 is used by default;

**IM** (identification means): a unique sequence of characters in machine-readable format, represented as a bar code, or stored in a RFID tag, or represented using another means (technology) of automatic identification, designated for the identification of instance of goods;

**LC** (labeling code): a unique sequence of characters used for forming identification means, generated by the labeling system operator;

**Marks**: document forms and documents with a certain level of protection having elements (means) of protection against forgery: unified control marks, protected physical data carriers, protection characters;

**PC** (protection character): a document with a certain level of protection used to protect against forgery of a physical data carrier of an identification means that does not have forgery protection elements (means);

**PPDC** (protected physical data carrier): a document form with a certain level of protection used for the application, storage and transmission of an identification means;

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

**SIS "Electronic Mark"**: state information system for labeling of goods with unified control marks or identification means; national component of the goods labeling system of an EAEU member state; a software and hardware complex providing receipt, processing, protection, storage and provision of received and processed data on labeling of goods with identification means, as well as information on operations performed with labeled goods;

**TLS** (Transport Layer Security): cryptographic protocols providing secure data transmission between nodes on the Internet;

**UCM** (unified control mark): a document with a certain level of protection used for labeling of goods;

**UNP**: Taxpayer's Number or other equivalent counterparty identifier.

## 2 GENERAL

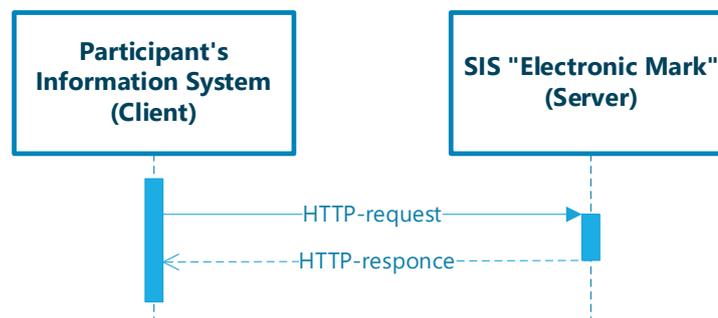
### 2.1 Cross-system coupling interface

This web service is designed to ensure coupling of accounting information systems of circulation participants with SIS "Electronic Mark" via API protocol, which is carried out within the REST client-server architecture over a secure HTTPS connection. The privacy of transmitted data is ensured by the TLS security protocol of version 1.2 or higher. Authentication of a circulation participant (user) within the current HTTP session is based on a unique API key (authentication token).

A general scheme of interaction and data exchange based on the "System-to-System" model via API protocol is shown in Figure 1.

The information system of the circulation participant is always the initiator of HTTP message exchange.

The information system of the circulation participant acts as a client and sends HTTP requests to SIS "Electronic Mark", which acts as a server. HTTP requests are transmitted between the client and the server. In response to an HTTP request, the server sends an HTTP response to the client.



**Figure 1** Scheme of cross-system coupling via API protocol

The HTTP request and HTTP response body is generated in JSON format by default.

Data is transmitted and processed in UTF-8 encoding.

## 2.2 HTTP request format

The following URLs provide access to the web service of SIS "Electronic Mark" via API protocol:

Commercial section, production circuit:

`https://api.datamark.by`

Test section, functional testing loop:

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

HTTP requests are sent to the web service using GET, POST, and PUT methods (request types) over a secure HTTPS connection.

In general, unless otherwise specified in the description of a specific API method, the HTTP request header (Headers) must contain the parameters specified in Table [2.1](#).

**Table 2.1** Default HTTP request header parameters

Key	Value
Content-Type	application/json; charset=utf-8
Content-Length	Request body length
Host	Web service URL address (commercial section or test section)
Token	Authentication token returned by API method /auth

## 2.3 HTTP response format

After receiving and processing an HTTP request, the web service returns an HTTP response:

with HTTP response code 200, in case of successful execution of the HTTP request; the HTTP response body is generated based on the structure specified in the description of the corresponding API method;

with HTTP response codes 4XX, in case of HTTP request execution with an error; HTTP response body in general case has parameters listed in Table [2.2](#).

**Table 2.2** HTTP response parameters, Code 4XX

Key	Required	Data type	Description
error	Yes	Integer	Error code
message	Yes	String	Message to the user and/or error description
details	No	Array	Detailed error description
Example of response body with description of request execution errors			
<pre>{   "error": 400,   "message": "Input parameters validation error",   "details": [     "Field ... must be ...",     "..."   ] }</pre>			

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

## 2.4 HTTP request execution procedure

In the active session, HTTP requests are sent in series. Each HTTP request calls the corresponding API method, which has the following characteristics described in Section 3 of this document:

- request type (call method);
- method name;
- set of passed and returned parameters;
- structure of the request and response body.

All API methods implemented in the web service are logically grouped into the following groups by their purpose:

- user identification methods (/users);
- product catalog handling methods (/catalogs, /items);
- order handling methods (/orders);
- code and label handling methods (/labels);
- report handling methods (/reports);
- agent handling methods (/contracts);
- shipment handling methods (/ships);
- service delivery report handling methods (/common);
- auxiliary methods (/downloads, /directories);

API methods of newer versions starting with v.2 have the prefix /vN at the beginning of the request string, where N is the version number.

**Attention!** Working with LC and/or AC requires using API methods of different versions separately, depending on the product group. Unless otherwise specified, in the description of API methods that are similar in functionality, the method selection of the corresponding version is defined as:

*representation A*, for v.1 and v.2 methods, applied only for the product groups "Dairy products" (group = milk), "Footwear" (group = shoes), "Tires" (group = tires), "Consumer goods" (group = clothes), "Perfume and eau de toilette" (group = perfum), "Cameras and flash lamps" (group = photo), "Packaged water" (group = water), "Non-alcoholic beverages and juices" (group = juice), "Residual stock. Footwear" (group = leftovers\_shoes), "Residual stock. Tires" (group = leftovers\_tires), "Residual stock. Consumer goods" (group = leftovers\_clothes), as well as UCM handling (group = ukz);

*representation B*, for v.3 methods, applied for the product groups "Dietary supplements" (group = dietary\_supplements), "Vegetable oils and fat-and-oil products" (group = oil\_fat), "Animal feed" (group = animal\_feed), "Preserves" (group = preserves), "Veterinary medicines" (group = veterinary\_medicines), "Sturgeon roe and salmon roe" (group = caviar), "Bicycles and bicycle frames" (group = bike), "Antiseptics and disinfectants" (group = antiseptics), "Food products (Groceries)" (group = grocery), "Cosmetics and household cleaning products" (group = cosmetics).

The API method call sequence depends on the subject area (product group, type of identification means, labeling methods) and the relevant business processes. Generalized flow charts of typical scenarios of using API methods in interaction with SIS "Electronic Mark" are provided in the following sections:

- for labeling of goods with IM, Annex [A](#);
- for labeling goods with UCM, Annex [B](#).

The specifics of performing specific operations of labeling business processes for different product groups are presented in Annex [C](#).

## 2.5 Data types and formats

The following source data types are used in this document:

- String: string value;
- Integer: integer value;
- Decimal: decimal value;
- Date: date and time;
- Boolean: Boolean value;
- Object: object;
- Array: data array.

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

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

## 2.6 Escaping

According to the JSON specification, invalid characters are subject to mandatory escaping.

The LC and AC may contain one or more non-printable delimiting control characters <GS> (Group Separator, ASCII character table code 29). Non-printable delimiting character value <GS> in JSON structures is passed in Unicode in escaped form as `\u001d`.

**Note:** In this document, in examples of HTTP request body or HTTP response body presented in JSON format, special delimiting characters <GS> are denoted as `\u001d`.

# 3 API METHODS

## 3.1 User identification methods

### 3.1.1 Method "Authorization"

- Request type: POST
- Request string: /auth
- Parameters:

**Table 3.1.1.1 HTTP request header parameters**

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

**Table 3.1.1.2 HTTP request parameters**

Key	Required	Data type	Description
username	Yes	String	User e-mail
password	Yes	String	User password
is_rules_agree	No	Boolean	Flag of having read and accepted the terms and conditions of the public contract on the provision of IM and services on provision and accounting of LC in SIS "Electronic Mark" (accession to the public contract published on the official website of the Operator): True: yes; False: no. <i>Note:</i> Specified once at the first authorization 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. <i>Notes:</i> 1. Returned if the user has read, agreed and accepted the terms of the public contract ( <i>is_rules_agree</i> = True). 2. Token validity period is 1 hour from the time of successful authorization. 3. The date and time of expiration is specified in the response in the <i>expires_in</i> requisite of the User object. 4. If the token expires, the user must re-authorize and get a new token.
token_refresh	String	Backup authentication token. <i>Note:</i> API is not used when working with methods.
user	Object	Authorized user data. User class object (see <a href="#">4.1.1</a> )
<b>Response body example</b>		
<pre>{   "token":   "eyJhbnRvaWQ0eXZlVZSjRHTjc1OXFTSUNnVjMyYTk0Y2ZkZGNQZjBiSDdBPStSImIzcyI6IlwvYXBpIiwibmJmIjoxNjg0OTEyODk4LCJleHAiOiJlYXN2aXRva0BiYnYuYnkiX0.GQz_nSTUCQJpHP313LQmBLiEPR80s05Cm-nIeJjhZjK",   "token_refresh":   "eyJhbnRvaWQ0eXZlVZSjRHTjc1OXFTSUNnVjMyYTk0Y2ZkZGNQZjBiSDdBPStSImIzcyI6IlwvYXBpIiwibmJmIjoxNjg0OTEyODk4LCJleHAiOiJlYXN2aXRva0BiYnYuYnkiX0.GQz_nSTUCQJpHP313LQmBLiEPR80s05Cm-nIeJjhZjK", }</pre>		

```

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": "С000 \"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 Method "Logout"

Request type: POST  
 Request string: /logout  
 Parameters: No input and output parameters

### 3.1.3 Method "Password change"

Request type: POST  
 Request string: /users/changePassword  
 Parameters:

**Table 3.1.3.1 HTTP request parameters**

Key	Required	Data type	Description
oldpassword	Yes	String	Old password
password	Yes	String	New password. <b>Note:</b> The password must meet all the following conditions: 1. At least 8 characters long; 2. Contains numbers, upper and lower case letters of the Latin alphabet, special characters; 3. Contains at least 1 capital letter, 1 number and 1 special character.
Request body example			
<pre>{   "oldpassword": "*****",</pre>			

```
"password": "*****"
}
```

**Table 3.1.3.2** HTTP response parameters, Code 200

Key	Data type	Description
status	String	Flag of successful completion of an operation
Response body example		
<pre>{   "status": "OK" }</pre>		

## 3.2 Product catalog handling methods

### 3.2.1 Method "Product catalog sections"

Request type: POST  
Request string: /catalogs/  
Parameters: No input parameters

**Table 3.2.1.1** HTTP response parameters, Code 200

Key	Data type	Description
catalogs	Array of Objects	Array with hierarchical structure (tree) of sections (categories) of product catalog. Category class objects (see <a href="#">4.1.3</a> )
Response body example (fragment)		
<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       }     }   ] }</pre>		

```
    },
    "code": 10081,
    "inner": [
      {
        "name": "Белье постельное, столовое, туалетное и кухонное",
        "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": 10082
      },
      ...
    ]
  },
  ...
  {
    "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 Method "Product category parameters"

Request type: POST

Request string: /catalogs/{:code}

Parameters:

**Table 3.2.2.1 HTTP request parameters**

Key	Required	Data type	Description
code	Yes	Integer	Code (identifier) of the section (category) of the product catalog, obtained from the /catalogs/ method (see <a href="#">3.2.1</a> ; the list of values is given in Table <a href="#">4.1.3.2</a> ).
page	No	Integer	Product list page number
<b>Request string example</b>			
/catalogs/10089			
<b>Request body example</b>			
<pre>{   "page": 1 }</pre>			

**Table 3.2.2.2 HTTP response parameters, Code 200**

Key	Data type	Description
	Object	Product catalog section (category). Category class object (see <a href="#">4.1.3</a> )
<b>Response body example (fragment)</b>		
<i>Option 1. Dairy products</i>		
<pre>{   "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         }       ]     }   ] }</pre>		

```
    },
    {
      "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": "Массовая доля жира, %",
      "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": "Дата изготовления",
        "value": null,
        "units": null,
        "code": 29,
        "is_required": true,
        "is_editable": false,
        "is_show_catalog ": false
      },
      {
        "type": "date",
        "name": "Дата 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": [
    ...
]
```

<b>Option 2. Goods labeled with UCM</b>
<pre>} {   "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": "за исключением 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",</pre>

```

        "is_my": false,
        "params": []
    }
]
}

```

### 3.2.3 Method "Item search"

Request type: POST  
Request string: /items/findItems  
Parameters:

**Table 3.2.3.1** HTTP request parameters

Key	Required	Data type	Description
search	Yes	String	Search value (GTIN or Item number/Model)
is_import	No	Boolean	Flag of search in the catalog of imported items: True: search among items imported from the catalogs of national operators of the EAEU member states; False: search among items registered by the circulation participants (default mode). <i>Note:</i> If the parameter is not specified, the default search is performed in <code>is_import = False</code> mode.
<b>Request body example</b>			
<pre> {   "search": "04814888492071" } </pre>			

**Table 3.2.3.2** HTTP response parameters, Code 200

Key	Data type	Description
items	Array of Objects	Array containing the descriptions of the items found. <code>Item</code> class objects (see <a href="#">4.1.4</a> )
<b>Response body example</b>		
<i>Option 1.</i> Default search ( <code>is_import = False</code> or not specified)		
<pre> {   "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"       }     }   ] } </pre>		

```
    },
    "document_id": null,
    "tnved": "4011100009",
    "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-Stock Company \"Belshina\"",
        "display_value": "Joint-Stock 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-Stock Company \"Belshina\"",
        "display_value": "Joint-Stock 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 imported items (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,
      "epass_sync": false,
      "operator_country": "643",
      "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 Method "Item addition by GTIN"

Request type: POST  
Request string: /items/addByGtin  
Parameters:

**Table 3.2.4.1 HTTP request parameters**

Key	Required	Data type	Description
gtin	Yes	String	Added item GTIN. <i>Template:</i> [0-9]{14}
Request body example			
<pre>{   "gtin": "04810268043116" }</pre>			

**Table 3.2.4.2 HTTP response parameters, Code 200**

Key	Data type	Description
	Object	Added item description. Item class object (see 4.1.4)
Response body example		
<pre>{   "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 паше, весовой, 8 кг ПФ в г. Барановичи",   "gtin": "04810268043116",   "gtin_check": 0,   "group": "milk",   "document_id": null,   "tnved": "0406909909",   "is_my": true,   "catalog": {     "code": 10089,     "name": "Молочная продукция"   },   "params": [     {       "name": "Идентификатор налогоплательщика (УНП/ИНН) производителя",       "units": null,       "code": 96,       "value": "200030514", </pre>		

```
        "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": "Массовая доля жира, %",
        "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 Method "GTIN registration status check"

Request type: POST  
Request string: /items/checkGtin  
Parameters:

Table 3.2.5.1 HTTP request parameters

Key	Required	Data type	Description
gtin_list	Yes	Array of Strings	List of GTINs of the checked items
{:gtin}	Yes	String	Item GTIN. <b>Template:</b> [0-9]{14}. <b>Limit:</b> Maximum 100 GTINs per request.
Request body example (fragment)			
<pre> {   "gtin_list": [     "04810347004489", </pre>			

```

    "04810347004960",
    ...
    "08430854527934"
  ]
}

```

**Table 3.2.5.2 HTTP response parameters, Code 200**

Key	Data type	Description
{:gtin}	Object	Check results for each GTIN value passed. Element key name corresponds to GTIN; element value contains the check result or error description
gtin_check	Boolean	Flag of the GTIN registration in the catalog: True: the item is registered; False: the item is pending registration
Response body example (fragment)		
<pre> {   "04810347004489": {     "gtin_check": true   },   "04810347004960": {     "gtin_check": false   },   ...   "08430854527934": {     "error": 404,     "message": "Not found"   } } </pre>		

### 3.2.6 Method "Item information change"

Request type: POST  
Request string: /items/edit  
Parameters:

**Table 3.2.6.1 HTTP request parameters**

Key	Required	Data type	Description
code	Yes	String	Internal code (identifier) of the product in the catalog. <b>Notes:</b> 1. The value of the code requisite of the Item class object is specified (see 4.1.4). 2. Available only for items added by a participant (i.e., having flag <code>is_my = True</code> ).
image	No	String	Product image in Base64 format. <b>Limits:</b> 1. Only JPG and PNG graphic files are accepted. 2. Maximum file size is 300 Kb. 3. The string must contain a reference to the

			MIME-type (data:image/jpeg;base64 or data:image/png;base64). <i>Note:</i> Passing a parameter with an empty string value ("image": "") will result in the deletion of the previously loaded image.
params	No	Array of Objects	Array of additional parameters describing the product. The set of parameters (additional characteristics) and conditions of their setting is determined by the belonging of the product to the corresponding section of the catalog. Param class objects (see <a href="#">4.1.5</a> ) listed in the attribute catalog_params of the Category class object (see <a href="#">4.1.3</a> ). <i>Note:</i> Available only for parameters for which value adjustment is enabled (i.e., is_editable = True).
code	Yes	Integer	Parameter code (identifier)
value	Yes	String, Integer, Decimal, Date, Boolean, Object	New parameter value, depends on its type
<b>Request body example</b>			
<pre>{   "code": "100020000000306",   "image": "data:image/jpeg;base64,/9j/4AAQSkZJRgABA ... 95x8ySorNoCD/2Q==",   "params": [     {       "code": 17,       "value": "КОЖА"     },     {       "code": 27,       "value": "Республика Беларусь"     }   ] }</pre>			

**Table 3.2.6.2** HTTP response parameters, Code 200

Key	Data type	Description
item	Object	Description of the product as modified and supplemented. Item class object (see <a href="#">4.1.4</a> )

*Note:* The method is only used for adding or changing (updating) the product image.

### 3.2.7 Method "Inventory"

Request type: POST  
Request string: /v2/items/inventory  
Parameters:

**Table 3.2.7.1** HTTP request parameters

Key	Required	Data type	Description
-----	----------	-----------	-------------

group	Yes	String	Product group (the list of values is given in Table 4.2.1.2)
name	Yes	String	Inventory number
date	Yes	Date	Inventory date. <b>Format:</b> YYYY-MM-DD
quantity	Yes	Integer	Total number of the residual stock based on inventory
<b>Request body example</b>			
<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. Inventory class object (see 4.1.9)
<b>Response body example</b>		
<pre>{   "id": "1259835",   "group": "shoes",   "name": "124",   "date": "2021-11-01",   "quantity": 3200 }</pre>		

**Note:** The method can be applied only from the date of enabling mandatory labeling and becomes unavailable after the end of the time period for transmitting information on the inventory of the residual stock established for the relevant product group.

### 3.2.8 Method "Inventory list"

Request type: POST

Request string: /v2/items/inventoryList

Parameters:

**Table 3.2.8.1** HTTP request parameters

Key	Required	Data type	Description
date_from	Yes	Date	Interval start date. <b>Format:</b> YYYY-MM-DD
date_to	Yes	Date	Interval end date. <b>Format:</b> YYYY-MM-DD
group	Yes	String	Product group (the list of values is given in Table 4.2.1.2)
page	Yes	Integer	Inventory list page number
<b>Request body example</b>			
<pre>{   "date_from": "2021-11-01",   "date_to": "2021-11-01", }</pre>			

```

"group": "shoes",
"page": 1
}
    
```

**Table 3.2.8.2** HTTP response parameters, Code 200

Key	Data type	Description
count	Integer	Total number of inventory in the sample with current selections (filters)
inventories_list	Array of Objects	Array containing information about inventory. Inventory class objects (see <a href="#">4.1.9</a> )
Response body example		
<pre> {   "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     }   ] }     </pre>		

### 3.3 Order handling methods

#### 3.3.1 Method "Code order"

##### 3.3.1.1. Representation A

Request type: POST  
Request string: /v2/orders/add  
Parameters:

**Table 3.3.1.1** HTTP request parameters

Key	Required	Data type	Description
label_type	Yes	Integer	Code type. Code ID retrieved from /labels/types method (see <a href="#">3.4.3</a> ; the list of values is given in Table <a href="#">4.2.2.2</a> ). <b>Limit:</b> Only codes with type label_type = 6 are available for order.
count	Yes	Integer	Number of codes to order. <b>Limit:</b> Maximum 1,000 codes per order.
comment	No	String	Order comment
Request body example			

```
{
  "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	Order details. Order class object (see <a href="#">4.1.6</a> )
Response body example		
<pre>{   "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     }   } }</pre>		

### 3.3.1.2. Representation B

Request type: POST  
Request string: /v3/orders/add  
Parameters:

**Table 3.3.1.3** HTTP request parameters

Key	Required	Data type	Description
label_type	Yes	Integer	Code type. Code ID retrieved from /labels/types method (see <a href="#">3.4.3</a> ; the list of values is given in Table <a href="#">4.2.2.2</a> ). <b>Limit:</b> Only codes with type label_type = 6 are available for order.
count	Yes	Integer	Number of codes to order. <b>Limit:</b> Maximum 1,000 codes per order.
comment	No	String	Order comment
Request body example			
<pre>{   "label_type": 6,   "count": 12,</pre>			

```
"comment": "Заказ кодов SSCC в количестве 12 (двенадцати) единиц"
}
```

**Table 3.3.1.4** HTTP response parameters, Code 200

Key	Data type	Description
id	Integer	Order number (identifier)
created_at	Date	Order creation date and time
Response body example		
<pre>{   "id": 2000489528,   "created_at": "2025-01-13 11:56:58" }</pre>		

### 3.3.2 Method "Order details"

#### 3.3.2.1. Representation A

Request type: POST  
Request string: /v2/orders  
Parameters:

**Table 3.3.2.1** HTTP request parameters

Key	Required	Data type	Description
id	Yes	String, Integer	Order number. Code ID obtained from /v2/orders/add (see <a href="#">3.3.1</a> ) and /v2/orders/addGroupOrders methods (see <a href="#">3.3.4.1</a> )
Request body example			
<pre>{   "id": "250879" }</pre>			

**Table 3.3.2.2** HTTP response parameters, Code 200

Key	Data type	Description
order	Object	Order details. Order class object (see <a href="#">4.1.6</a> )
Response body example		
<pre>{   "order": {     "id": 250879,     "created_at": "2020-12-15 10:03:00+03",     "updated_at": "2020-12-15 10:02:25+03",     "count": 20,     "comment": "Заказ 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",   } }</pre>		

```

        "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\u001d93V1TU",
        "0104810268024894212w2RoVr0\u001d93!Y+!",
        "0104810268024894212cVXPq8R\u001d93w=Zo",
        ...
        "01048102680248942125AYX1Qy\u001d93Mk_R"
    ]
}
}

```

### 3.3.2.2. Representation B

Request type: GET  
 Request string: /v3/orders/list/{:id}  
 Parameters:

**Table 3.3.2.3 HTTP request parameters**

Key	Required	Data type	Description
id	Yes	Integer	Order number. Order ID, obtained from methods /v3/orders/add (see <a href="#">3.3.1.2</a> ), /v3/orders/addGroupOrders (see <a href="#">3.3.4.2</a> ) or /v3/labels/addAggregate (see <a href="#">3.4.1.2</a> )
<b>Request string example</b>			
/v3/orders/list/2000488178			

**Table 3.3.2.4 HTTP response parameters, Code 200**

Key	Data type	Description
detail	Object	Order details. Order class object (see <a href="#">4.1.6</a> )
<b>Response body example</b>		

**Option 1.** Information on ordering of labeling codes

```
{
  "detail": {
    "id": 2000488178,
    "created_at": "2024-11-29T20:25:26.699534",
    "updated_at": "2024-11-30T13:00:14.554352",
    "completed_at": "2024-11-30T13:00:14.554352",
    "count": 5,
    "comment": "Длинный КМ",
    "gtin": {
      "name": "Биологически активные добавки (БАД)",
      "articul": "№1",
      "gtin": "04812845008808"
    },
    "group": {
      "code": "dietary_supplements",
      "name": "Биологически активные добавки к пище"
    },
    "type": {
      "id": 7,
      "name": "Код стран ЕАЭС идентификации товара"
    },
    "files": {
      "labels": {
        "filename": "507-2024-11-29-21-5-15-jIZvHp",
        "downloads": 0
      }
    },
    "user": {
      "name": "Валерий",
      "lastname": "Матоха",
      "email": "matohasvitok@bbv.by"
    },
    "user_info": {
      "name": "СООО \"Свиток\"",
      "address": "г. Минск, ул. Тимирязева, д.2, ком. 16",
      "unp": "190704463",
      "gln": "4811612900000",
      "country": "112",
      "is_rules_agree": true
    },
    "status": {
      "code": 30,
      "message": "Выполнен"
    }
  }
}
```

**Option 2.** Information on ordering of aggregation codes (in the SSCC format)

```
{
  "detail": {
    "id": 2000489528,
    "created_at": "2025-01-13T11:56:58.242603",
    "updated_at": "2025-01-13T11:56:58.535544",
    "completed_at": "2025-01-13T11:56:58.535544",
    "count": 12,
    "comment": "Заказ кодов SSCC в количестве 12 (двенадцати) единиц",
    "type": {
```

```

        "id": 6,
        "name": "Код агрегации (SSCC)"
    },
    "files": {
        "labels": {
            "filename": "507-2025-1-13-11-56-58-ntJXgE",
            "downloads": 1
        }
    },
    "user": {
        "name": "Валерий",
        "lastname": "Матоха",
        "email": "matohasvitok@bbv.by"
    },
    "user_info": {
        "name": "С000 \Свиток\\"",
        "address": "г. Минск, ул. Тимирязева, д.2, ком. 16",
        "unp": "90704463",
        "gln": "4811612900000",
        "country": "112",
        "is_rules_agree": true
    },
    "status": {
        "code": 30,
        "message": "Выполнен"
    }
}
}
}

```

**Option 3.** Information on ordering addition of custom aggregation codes

```

{
    "detail": {
        "id": 2000489512,
        "created_at": "2025-01-13T10:00:57.734920",
        "updated_at": "2025-01-13T10:00:57.845818",
        "completed_at": "2025-01-13T10:00:57.845822",
        "count": 6,
        "comment": null,
        "type": {
            "id": 13,
            "name": "Код агрегации(импорт)"
        },
    },
    "files": {
        "labels": {
            "filename": "507-2025-1-13-10-0-57-tclzov",
            "downloads": 1
        },
        "labels_error": {
            "filename": "507-2025-1-13-10-0-57-tclzov_bad_aggregates",
            "downloads": 1
        }
    },
    "user": {
        "name": "Валерий",
        "lastname": "Матоха",
        "email": "matohasvitok@bbv.by"
    },
    "user_info": {

```

```

        "name": "COOO \\"Свиток\\"",
        "address": "г. Минск, ул. Тимирязева, д.2, ком. 16",
        "unp": "190704463",
        "gln": "4811612900000",
        "country": "112",
        "is_rules_agree": true
    },
    "status": {
        "code": 30,
        "message": "Выполнен"
    }
}
}
}

```

### 3.3.3 Method "Order list"

#### 3.3.3.1. Representation A

Request type: POST  
Request string: /orders/list  
Parameters:

**Table 3.3.3.1 HTTP request parameters**

Key	Required	Data type	Description
date_from	No	Date	Interval start date. <b>Format:</b> YYYY-MM-DD
date_to	No	Date	Interval end date. <b>Format:</b> 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 retrieved from /labels/types method (see <a href="#">3.4.3</a> ; the list of values is given in Table <a href="#">4.2.2.2</a> )
page	Yes	Integer	Order list page number
Request body example			
<pre> {   "date_from": "2020-11-11",   "date_to": "2020-11-11",   "label_type": 6,   "page": 1 } </pre>			

**Table 3.3.3.2 HTTP response parameters, Code 200**

Key	Data type	Description
count	Integer	Total number of orders in the sample with current selections (filters)
orders_list	Array of Objects	Array containing information about orders. Order class objects (see <a href="#">4.1.6</a> )
Response body example (fragment)		
<pre> {   "count": 9,   "orders_list": [ </pre>		

```
{
  {
    "id": 253170,
    "created_at": "2020-11-11 17:21:13+03",
    "updated_at": "2020-11-11 17:21:14+03",
    "count": 1,
    "comment": "Из Postman Коды агрегации (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": {
      "name": "Валерий",
      "lastname": "Матоха",
      "info": {
        "name": "С000 \"Свиток\"",
        "address": "г. Минск, ул. Тимирязева, д.2, ком. 16",
        "unp": 190704463,
        "gln": "4811612900000",
        "country": "112",
        "is_rules_agree": true
      }
    }
  },
  ...
]
}
```

### 3.3.3.2. Representation B

Request type: GET  
Request string: /v3/orders/list?[page={:page}]  
&[page\_size={:page\_size}]  
&[ordering={:ordering}]  
&[start\_date={:start\_date}]  
&[end\_date={:end\_date}]  
&[start\_completed\_at={:start\_completed\_at}]  
&[end\_completed\_at={:end\_completed\_at}]

&[type\_id={:type\_id}]  
&[gtin={:gtin}]  
&[search={:search}]

Parameters:

**Table 3.3.3.3 HTTP request parameters**

Key	Required	Data type	Description
page	No	Integer	Order list page number
page_size	No	Integer	Number of entries per page. <i>Note:</i> Maximum 200, by default, up to 15 per page.
ordering	No	String	Data sorting conditions: <b>created_at:</b> by creation date and time in ascending order; - <b>created_at:</b> by creation date and time in descending order; <b>id:</b> by ascending order number (identifier); - <b>id:</b> by descending order number (identifier)
start_date	No	Date	Interval start date. <i>Format:</i> YYYY-MM-DD
end_date	No	Date	Interval end date. <i>Format:</i> YYYY-MM-DD
start_completed_at	No	Date	Order completion start date. <i>Format:</i> YYYY-MM-DD <i>Note:</i> Orders completed after the specified date are retrieved.
end_completed_at	No	Date	Order completion end date. <i>Format:</i> YYYY-MM-DD <i>Note:</i> Orders completed before the specified date are retrieved.
gtin	No	String	GTIN of the item for which the LC order is generated
type_id	No	Integer	Code type. Code ID retrieved from /labels/types method (see <a href="#">3.4.3</a> ; the list of values is given in Table <a href="#">4.2.2.2</a> )
search	No	Integer	Order number. Order ID by which the list is searched
<b>Request string example</b>			
<i>Option 1.</i> Selection by parameters			
/v3/orders/list?page=2&ordering=-created_at&start_date=2024-11-29&end_date=2024-11-29&type_id=7			
<i>Option 2.</i> Search by order number			
/v3/orders/list?search=2000488178			

**Table 3.3.3.4 HTTP response parameters, Code 200**

Key	Data type	Description
next	String	Link to the next page of the order list. <i>Note:</i> Null if none.
previous	String	Link to the previous page of the order list.

		<i>Note:</i> Null if none.
results	Array of Objects	Array containing information about orders. Order class objects (see <a href="#">4.1.6</a> )
<b>Response body example (fragment)</b>		
<i>Option 1.</i> Selection by parameters		
<pre> {   "next": null,   "previous": "http://api.datamark.by/v3/orders/list?end_date=2024-11-29&amp;start_date=2024-11-29",   "results": [     ...     {       "id": 2000488179,       "created_at": "2024-11-29T20:25:26.699655",       "updated_at": "2024-11-30T13:00:31.219103",       "user_id": 507,       "user_info_id": 129,       "count": 5,       "comment": "Короткий КМ",       "gtin": "00625273050882",       "completed_at": "2024-11-30T13:00:31.219103",       "type": {         "id": 7,         "name": "Код стран ЕАЭС идентификации товара"       },       "files": {         "labels": {           "filename": "507-2024-11-29-21-5-15-RWSyGE",           "downloads": 0         }       },       "user": {         "name": "Валерий",         "lastname": "Матуха",         "email": "matohasvitok@bbv.by"       },       "user_info": {         "name": "C000 \"Свиток\"",         "address": "г. Минск, ул. Тимирязева, д.2, ком. 16",         "unp": "190704463",         "gln": "4811612900000",         "country": "112",         "is_rules_agree": true       },       "status": {         "code": 30,         "message": "Выполнен"       },       "item": {         "name": "БИОЛОГИЧЕСКИ АКТИВНАЯ ДОБАВКА",         "articul": "",         "gtin": "00625273050882"       },       "group": {         "code": "dietary_supplements",         "name": "Биологически активные добавки к пище"       }     }   ] } </pre>		

```
    }
  },
  {
    "id": 2000488178,
    "created_at": "2024-11-29T20:25:26.699534",
    "updated_at": "2024-11-30T13:00:14.554352",
    "user_id": 507,
    "user_info_id": 129,
    "count": 5,
    "comment": "Длинный КМ",
    "gtin": "04812845008808",
    "completed_at": "2024-11-30T13:00:14.554352",
    "type": {
      "id": 7,
      "name": "Код стран ЕАЭС идентификации товара"
    },
    "files": {
      "labels": {
        "filename": "507-2024-11-29-21-5-15-jIZvHr",
        "downloads": 0
      }
    },
    "user": {
      "name": "Валерий",
      "lastname": "Матюха",
      "email": "matohasvitok@bbv.by"
    },
    "user_info": {
      "name": "С000 \"Свиток\"",
      "address": "г. Минск, ул. Тимирязева, д.2, ком. 16",
      "unp": "190704463",
      "gln": "4811612900000",
      "country": "112",
      "is_rules_agree": true
    },
    "status": {
      "code": 30,
      "message": "Выполнен"
    },
    "item": {
      "name": "Биологически активные добавки (БАД)",
      "articul": "№1",
      "gtin": "04812845008808"
    },
    "group": {
      "code": "dietary_supplements",
      "name": "Биологически активные добавки к пище"
    }
  },
  ...
]
}
```

**Option 2. Search by order number**

```
{
  "next": null,
  "previous": null,
  "results": [
```

```
{
  "id": 2000488178,
  "created_at": "2024-11-29T20:25:26.699534",
  "updated_at": "2024-11-30T13:00:14.554352",
  "user_id": 507,
  "user_info_id": 129,
  "count": 5,
  "comment": "Длинный КМ",
  "gtin": "04812845008808",
  "completed_at": "2024-11-30T13:00:14.554352",
  "type": {
    "id": 7,
    "name": "Код стран ЕАЭС идентификации товара"
  },
  "files": {
    "labels": {
      "filename": "507-2024-11-29-21-5-15-jIZvHр",
      "downloads": 0
    }
  },
  "user": {
    "name": "Валерий",
    "lastname": "Матуха",
    "email": "matohasvitok@bbv.by"
  },
  "user_info": {
    "name": "Предприятие РБ для тестирования",
    "address": "г. Минск, ул. Тимирязева, д.2, ком. 16",
    "unp": "190704463",
    "gln": "4811612900000",
    "country": "112",
    "is_rules_agree": true
  },
  "status": {
    "code": 30,
    "message": "Выполнен"
  },
  "item": {
    "name": "Биологически активные добавки (БАД)",
    "articul": "№1",
    "gtin": "04812845008808"
  },
  "group": {
    "code": "dietary_supplements",
    "name": "Биологически активные добавки к пище"
  }
}
]
```

### 3.3.4 Method "Group order of labeling codes"

#### 3.3.4.1. Representation A

Request type: POST  
Request string: /v2/orders/addGroupOrders

Parameters:

**Table 3.3.4.1 HTTP request parameters**

Key	Required	Data type	Description
group	Yes	String	Product group (the list of values is given in Table 4.2.1.2)
orders	Yes	Object	LC order parameters
{:gtin}	Yes	Object	The element key name corresponds to the GTIN value of the item. The element value contains order parameters to the specified GTIN. <b>Template:</b> [0-9]{14}. <b>Limit:</b> Maximum 100 GTINs per request.
count	Yes	Integer	Number of codes to order. <b>Limit:</b> For each GTIN in the request depending on the type of code: 1. For type label_type = 12: maximum 100,000 codes; 2. For other types: maximum 10,000 codes.
label_type	Yes	Integer	Code type. Code ID retrieved from /labels/types method (see 3.4.3; the list of values is given in Table 4.2.2.2)
comment	No	String	Order comment
typography_id	No	Integer	Printing house ID obtained by /users/typographyList method (see 3.8.4)
typography_doc	No	String	Number and date of the order (printing house contract)
<b>Request body example (fragment)</b>			
<pre>{   "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"     },     ...   } }</pre>			

**Table 3.3.4.2** HTTP response parameters, Code 200

Key	Data type	Description
{:gtin}	Object	A list of order identifiers for each GTIN value provided. Element key name corresponds to the GTIN of the item. The element value contains order identifiers.
id	Integer	Order number (identifier)
created_at	Date	Order creation date and time
Response body example (fragment)		
<pre>{   "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"   },   ... }</pre>		

### 3.3.4.2. Representation B

Request type: POST

Request string: /v3/orders/addGroupOrders

Parameters:

**Table 3.3.4.3** HTTP request parameters

Key	Required	Data type	Description
group	Yes	String	Product group (the list of values is given in Table 4.2.1.2)
orders	Yes	Object	LC order parameters
{:gtin}	Yes	Object	The element key name corresponds to the GTIN value of the item. The element value contains order parameters to the specified GTIN. <b>Template:</b> [0-9]{14}. <b>Limit:</b> Maximum 100 GTINs per request.
count	Yes	Integer	Number of codes to order. <b>Limit:</b> Maximum 10,000 codes per each GTIN in the request
label_type	Yes	Integer	Code type. Code ID retrieved from /labels/types method (see 3.4.3; the list of values is given in Table 4.2.2.2). <b>Limit:</b> Only codes with type label_type = 7 are available for order
template_id	No	Integer	LC template. LC structure ID defining the option of data group composition set for the corresponding product group (the list of values is given in Annex E).

			<b>Note:</b> By default, the basic (recommended) template set for the corresponding product group is used.
comment	No	String	Order comment
typography_id	No	Integer	Printing house ID obtained by /users/typographyList method (see <a href="#">3.8.4</a> )
typography_doc	No	String	Number and date of the order (printing house contract)
<b>Request body example (fragment)</b>			
<pre>{   "group": "dietary_supplements",   "orders": {     "04812845008808": {       "count": 5,       "label_type": 7,       "comment": "Длинный КМ"     },     "00625273050882": {       "count": 5,       "label_type": 7,       "template_id": 23,       "comment": "Короткий КМ"     },     ...   } }</pre>			

**Table 3.3.4.4** HTTP response parameters, Code 200

Key	Data type	Description
orders	Object	LC order identifier list
{:gtin}	Object	The element key name corresponds to the GTIN value of the item. The element value contains order details by the specified GTIN.
id	Integer	Order number (identifier)
created_at	Date	Order creation date and time
<b>Response body example (fragment)</b>		
<pre>{   "orders": {     "04812845008808": {       "id": 2000488178,       "created_at": "2024-11-29 20:25:26"     },     "00625273050882": {       "id": 2000488179,       "created_at": "2024-11-29 20:25:26"     },     ...   } }</pre>		

### 3.3.5 Method "Order status list"

Request type: POST

Request string: /v2/orders/statusList  
Parameters:

**Table 3.3.5.1 HTTP request parameters**

Key	Required	Data type	Description
label_type	Yes	Integer	Code type. Code ID retrieved from /labels/types method (see <a href="#">3.4.3</a> ; the list of values is given in Table <a href="#">4.2.2.2</a> )
page	Yes	Integer	Order list page number
Request body example			
<pre>{   "label_type": 12,   "page": 1 }</pre>			

**Table 3.3.5.2 HTTP response parameters, Code 200**

Key	Data type	Description
orders_list	Array of Objects	Array containing brief information about orders and their current statuses. Order class objects (see <a href="#">4.1.6</a> )
Response body example (fragment)		
<pre>{   "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": {</pre>		

```

        "id": 12,
        "name": "Код РБ идентификации молочной продукции",
        "method": "nobsomilk",
        "digits_count": null,
        "is_bso": false,
        "letters_count": null,
        "is_autotake": false
    },
    ...
]
}

```

### 3.3.6 Method "Number of completed orders"

Request type: POST  
Request string: /orders/count  
Parameters:

**Table 3.3.6.1** HTTP request parameters

Key	Required	Data type	Description
date_from	No	Date	Interval start date. <b>Format:</b> YYYY-MM-DD
date_to	No	Date	Interval end date. <b>Format:</b> 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 retrieved from /labels/types method (see <a href="#">3.4.3</a> ; the list of values is given in Table <a href="#">4.2.2.2</a> )
page	Yes	Integer	Order list page number
Request body example			
<pre> {   "date_from": "2021-05-28",   "date_to": "2021-05-30",   "label_type": 12,   "page": 1 } </pre>			

**Table 3.3.6.2** HTTP response parameters, Code 200

Key	Data type	Description
count	Integer	Total number of completed orders in the sample with current selections (filters)
Response body example		
<pre> {   "count": 9 } </pre>		

**Note:** The data returned by /orders/count method is updated not sooner than 2 (two) minutes after the last generation of current data.

### 3.3.7 Method "Ordered code list"

Request type: POST  
Request string: /v3/orders/downloads  
Parameters:

**Table 3.3.7.1 HTTP request parameters**

Key	Required	Data type	Description
filename	Yes	String	File name specified in the same-named parameter of the files section of the Order class object (see 4.1.6) returned by /v3/orders/list/{:id} (see 3.3.2.2) or /v3/orders/list method (see 3.3.3.2)
<b>Request body example</b>			
<pre>{   "filename": "507-2024-11-29-21-5-15-jIZvHp" }</pre>			

**Table 3.3.7.2 HTTP response parameters, Code 200**

Key	Data type	Description
labels	Array of Strings	Array (list) of LCs from the order
<b>Response body example (fragment)</b>		
<pre>{   "labels": [     "0104812845008808215CjBmJL,pugAq\u001d91FFD0\u001d92dGVzdBSSkDJP1g/BAA     bCv3wv/OaXx8Fb2u5JT9pZgKc=",     "0104812845008808215&amp;)' )WXeauGKm\u001d91FFD0\u001d92dGVzdPBTdSpuGymZRA     LebENUimTf271ALnTtHAC/ZFY=",     ...     "0104812845008808215j.FUYPHZM_nV\u001d91FFD0\u001d92dGVzdBs1wK4e8qW5oR     XvEE8W49YCi+uoC4R3REf5730="   ] }</pre>		

## 3.4 Code and label handling methods

### 3.4.1 Method "Custom aggregation code addition"

#### 3.4.1.1. Representation A

Request type: POST  
Request string: /v2/labels/addAggregate  
Parameters:

**Table 3.4.1.1 HTTP request parameters**

Key	Required	Data type	Description
aggregate	Yes	Array of Strings	Array (list) of ACs created by the circulation participant. <b>Limits:</b> 1. AC length: 21 to 74 characters. 2. Maximum 1,000 ACs per request. <b>Note:</b> The code ordering method

			/v2/orders/add with code type label_type = 6 is used to generate AC in SSCC format.
<b>Request body example (fragment)</b>			
<pre>{   "aggregate": [     "000481901888000000320204811309016724376\u001d1320111110abcdef",     "000481901888000000180204811309016724376\u001d1320111230987zyx",     ...     "000481901888000000560204811309016724376\u001d1320111230987d1e"   ] }</pre>			

**Table 3.4.1.2** HTTP response parameters, Code 200

Key	Data type	Description
count	Object	Number of ACs (total):
success	Integer	- successfully added;
failed	Integer	- incorrect (rejected)
error	Array of Strings	Array (list) of rejected ACs with error descriptions
<b>Response body example</b>		
<pre>{   "count": {     "success": 98,     "failed": 2   },   "error": [     "010481533023884121wbv2 already exists",     "8974klsakbdlalk3 already exists"   ] }</pre>		

### 3.4.1.2. Representation B

Request type: POST  
 Request string: /v3/labels/addAggregate  
 Parameters:

**Table 3.4.1.3** HTTP request parameters

Key	Required	Data type	Description
aggregate	Yes	Array of Strings	Array (list) of ACs created by the circulation participant. <b>Limits:</b> 1. AC length: 21 to 74 characters. 2. Maximum 1,000 ACs per request. <b>Notes:</b> 1. The AC is checked for uniqueness in SIS "Electronic Mark". 2. The code ordering method /v3/orders/add with code type label_type = 6 is used to generate AC in SSCC format.

Request body example (fragment)
<pre>{   "aggregate": [     "1323010520222111\u001d4144811612900000422112",     "1323010520222112\u001d4144811612900000422112",     ...     "1323010520222119\u001d4144811612900000422112"   ] }</pre>

**Table 3.4.1.4** HTTP response parameters, Code 200

Key	Data type	Description
id	Integer	Order number (identifier)
created_at	Date	Order creation date and time
success	Integer	Number of successfully added ACs
failed	Integer	Number of incorrect (rejected) ACs
Response body example		
<pre>{   "id": 2000489512,   "created_at": "2025-01-13 10:00:57",   "success": 6,   "failed": 3 }</pre>		

**Note:** Custom ACs generated and submitted by circulation participants are automatically assigned with the code type `label_type = 13`.

### 3.4.2 Method "Deletion of codes from aggregation"

Request type: POST  
Request string: /labels/aggregateDelete  
Parameters:

**Table 3.4.2.1** HTTP request parameters

Key	Required	Data type	Description
aggregate	Yes	String	AC (SSCC or custom)
labels	Yes	Array of Strings	Array of lower-level LC or AC excluded from aggregation. <b>Limit:</b> Codes must be of the same or uniform types. Simultaneous use of lower-level LCs and ACs at the same grouping level is not allowed
Request body example			
<pre>{   "aggregate": "000481901888000000180204811309016724376\u001d1320111230987zyx",   "labels": [     "0104811309016724212R7V7577\u001d93iv4=",     "01048113090167242128v8LZVn\u001d936rX0"   ] }</pre>			

**Table 3.4.2.2** HTTP response parameters, Code 200

Key	Data type	Description
aggregate	String	Processed AC
count	Object	Number of codes (total):
total	Integer	- remaining in the aggregate;
deleted	Integer	- excluded from the aggregate;
failed	Integer	- incorrect
details	Array of Strings	Incorrect code array
Response body example		
<pre>{   "aggregate": "000481901888000000180204811309016724376\u001d1320111230987zyx",   "count": {     "deleted": 2,     "total": 4,     "failed": 0   } }</pre>		

### 3.4.3 Method "Types of codes or labels"

Request type: GET  
Request string: /labels/types  
Parameters: No input parameters

**Table 3.4.3.1** HTTP response parameters, Code 200

Key	Data type	Description
types_list	Array of Objects	Code type array LabelType class objects (see 4.2.2)
Response body example (fragment)		
<pre>{   "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     },     ...   ] }</pre>		

```

        "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
    }
]
}

```

### 3.4.4 Method "Discarding codes or labels"

#### 3.4.4.1. Option 1

It is used only to send information about the PCs used for labeling goods.

Request type: POST  
Request string: /labels/destroy  
Parameters:

**Table 3.4.4.1** HTTP request parameters

Key	Required	Data type	Description
shipping_doc	No	String	Type of document confirming the fact of discarding. Enumeration value code shippingDocs obtained by /directories/{:code} method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.3</a> )
nomer_doc	No	String	Number of document confirming the fact of discarding
reason	Yes	String	Discarding reason. Enumeration value code reasons obtained by /directories/{:code} method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.4</a> ). <b>Limit:</b> Only discard of PCs used for marking goods (reason = usemark)
item_name	Yes	String	Item name
period	Yes	String	The period (quarter) during which the forms were used
comment	No	String	Comment to the discarding operation

labels	Yes	Array of Strings	Array series and numbers of forms (labels) to be discarded. <b>Limit:</b> Only series and numbers of PCs (code type <code>label_type = 19</code> ) are listed
<b>Request body example (fragment)</b>			
<pre>{   "reason": "usemark",   "item_name": "Обувь мужская",   "period": "IV квартал 2021 г.",   "comment": "Использование знаков защиты",   "labels": [     "AA012300001",     "AA012300002",     "AA012300003",     ...     "AA012301000"   ] }</pre>			

**Table 3.4.4.2** HTTP response parameters, Code 200

Key	Data type	Description
message	String	Information message
count	Integer	Number of used PCs
<b>Response body example</b>		
<pre>{   "message": "Успешно обработано",   "count": 1000 }</pre>		

#### 3.4.4.2. Option 2

It is used to discarding codes (LCs or ACs) or labels (UCMs, PPDCs or PCs) for reasons other than their use for labeling goods (for example, damage, loss, theft, by decision of a business entity or other reasons).

Request type: POST  
Request string: `/v3/labels/destroy`  
Parameters:

**Table 3.4.4.3** HTTP request parameters

Key	Required	Data type	Description
transaction_date	Yes	Date	Date of operation. <b>Format:</b> YYYY-MM-DD
label_type	Yes	Integer	Type of codes or labels to be discarded. Code ID retrieved from <code>/labels/types</code> method (see <a href="#">3.4.3</a> ; the list of values is given in Table <a href="#">4.2.2.2</a> )
reason	Yes	String	Discarding reason. Enumeration value code reasons obtained by <code>/directories/{:code}</code> method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.4</a> ).

			<b>Note:</b> Except for "Use (labeling)" (reason = usemark) – selection of the specified discarding reason is not allowed
document_type	Yes	String	Type of underlying document confirming the fact of discarding. Enumeration value code shippingDocs obtained by /directories/{:code} method (see 3.8.3; the list of values is given in Table 4.3.3)
document_date	Yes	Date	Date of underlying document confirming the fact of discarding. <b>Format:</b> YYYY-MM-DD
document_number	Yes	String	Number of underlying document confirming the fact of discarding
comment	No	String	Comment to the discarding operation
labels	Yes/No	Array of Strings	Array of LCs or series and numbers of UCMs, PCs or PPDCs to be discarded. <b>Limit:</b> Maximum 100,000 codes per request. <b>Note:</b> Must be set if aggregates is not specified
aggregates	Yes/No	Array of Strings	Array of ACs to be discarded. <b>Notes:</b> 1. Must be set if label is not specified. 2. When discarding ACs, all nested LCs or ACs, including those at lower nesting levels, are also discarded.
<b>Request body example (fragment)</b>			
<pre>{   "transaction_date": "2025-01-10",   "label_type": 7,   "reason": "damage",   "document_type": "akt",   "document_date": "2025-01-12",   "document_number": "2a",   "labels": [     "0100625273050882215&lt;AID7N0Td05T\u001d91FFD0\u001d92dGVzdBUtb1+pU+1oaj     FxrtPOKkvMoaRYA0cQyD+IWec=",     "0100625273050882215JKLDGHJT(34Q\u001d91FFD0\u001d92dGVzd06uV4PpGZGxDq     ZMgE++H9WeH7vMnBvv1MGCHg=",     ...     "0100625273050882215XmFKi)t.pqB5\u001d91FFD0\u001d92dGVzdPSQ8VqE0fx8Ua     rPuutnIL6Y2LvykFo8w9gWP14="   ] }</pre>			

**Note:** The request must always contain at least one of the parameters in which the codes are listed ("labels" for LCs, series and numbers of UCMs, PCs or PPDCs or "aggregates" for ACs).

**Table 3.4.4.4 HTTP response parameters, Code 200**

Key	Data type	Description
report_id	Integer	Report ID

report_uid	String	Unique identifier (UID) of the label discard report (report_type = 11)
<b>Response body example</b>		
<pre>{   "report_id": 5443,   "report_uid": "0d3e86d3-ddf0-4664-a612-4fa9130bd7df" }</pre>		

### 3.4.5 Method "Discarding list"

Request type: POST  
Request string: /stats/destroyList  
Parameters:

**Table 3.4.5.1 HTTP request parameters**

Key	Required	Data type	Description
page	Yes	Integer	Discarding list page number
reason	No	Array of Strings	Discarding reason array. Enumeration value codes reasons obtained by /directories/{:code} method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.4</a> )
<b>Request body example</b>			
<pre>{   "page": 1,   "reason": [     "damage",     "lost",     "usemark"   ] }</pre>			

**Table 3.4.5.2 HTTP response parameters, Code 200**

Key	Data type	Description
count	Integer	Total number of discardings in the sample with current selections (filters)
destroy_list	Array of Objects	Array of discarded LCs, ACs or series and numbers of UCM forms
<b>Response body example</b>		
<pre>{   "count": 302,   "destroy_list": [     {       "created_at": "2021-06-17 16:28:41+03",       "message": {         "name": "ФН254",         "reason": "damage",         "message": "",         "shipping_doc": "akt"       },       "user": {         "name": "Александр",         "lastname": "Дешиц", </pre>		

```
        "info": {
            "name": "Индивидуальный предприниматель Дешиц Александр
Евгеньевич",
            "address": "220116, г Минск, проспект \" Газеты Звезда\" 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": "019481900857000421011050791AAA"
},
{
    "created_at": "2021-06-29 16:37:27+03",
    "message": {
        "name": "C152-54",
        "reason": "lost",
        "message": "",
        "shipping_doc": "akt"
    },
    "user": {
        "name": "Александр",
        "lastname": "Дешиц",
        "info": {
            "name": "Индивидуальный предприниматель Дешиц Александр
Евгеньевич",
            "address": "220116, г Минск, проспект \" Газеты Звезда\" 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": "019481900857000421011050812AAA"
},
{
    "created_at": "2021-06-29 16:38:29+03",
    "message": {
        "name": "AA25487",
        "reason": "usemark",
        "message": "",
        "shipping_doc": "akt"
    },
    "user": {
        "name": "Александр",
        "lastname": "Дешиц",
        "info": {
            "name": "Индивидуальный предприниматель Дешиц Александр
Евгеньевич",
            "address": "220116, г Минск, проспект \" Газеты Звезда\" 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": "019481900857000421011050813AAA"
}
]
}

```

### 3.4.6 Method "Labeling code validation"

Request type: POST  
Request string: /v2/labels  
Parameters:

Table 3.4.6.1 HTTP request parameters

Key	Required	Data type	Description
labels	Yes	Array of Strings	Array of LCs to be checked <i>Limit:</i> Maximum 1,000 LCs per request
<b>Request body example (fragment)</b>			

```
{
  "labels": [
    "0104811215019130212tSKmLjdc6N9\u001d91TEST\u001d92testtest...
testtest",
    "01048112150191302127USNfqUza15o\u001d91TEST\u001d92testtest...
testtest",
    ...
    "0104811215019130212K2HSsx7xaPPj\u001d91TEST\u001d92testtest...
testtest"
  ]
}
```

**Table 3.4.6.2** HTTP response parameters, Code 200

Key	Data type	Description
labels	Array of Objects	Array containing check details for each LC
label	String	LC to be checked
found	Boolean	Flag (confirmation) of LC availability in the information system: True: confirmed; False: not confirmed
status	Object	Current LC status LabelStatus class object (see 4.2.3). <i>Note:</i> Returned if found = true.
is_my	Boolean	Code assignment flag: True: to the current circulation participant; False: to another agent. <i>Note:</i> Returned if found = true.
type	Object	Code type. LabelType class object (see 4.2.2). <i>Note:</i> Returned if found = true.

**Response body example (fragment)**

```
"labels": [
  {
    "label": "0104811215019130212tSKmLjdc6N9\u001d91TEST\u001d92test
test... 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\u001d91TEST\u001d92test
test...test",
    "found": false
  },
  ...
  {
    "label": "0104811215019130212K2HSsx7xaPPj\u001d91TEST\u001d92test
test... test",
```

```

    "found": true,
    "status": {
      "code": 50,
      "message": "Промаркирован"
    }
    "is_my": false,
    "type": {
      "id": 20,
      "name": "Код РБ идентификации товара (обувь, шины, легкая
промышленность)"
    }
  }
]

```

### 3.4.7 Method "Information on the aggregate structure and composition"

#### 3.4.7.1. Representation A

Request type: POST  
Request string: /labels/aggregateInfo  
Parameters:

**Table 3.4.7.1 HTTP request parameters**

Key	Required	Data type	Description
aggregate	Yes	String	AC (SSCC or custom)
Request body example			
<pre>{   "aggregate": "00048116129001002492" }</pre>			

**Table 3.4.7.2 HTTP response parameters, Code 200**

Key	Data type	Description
	Array of Objects	Aggregate composition. <i>Note.</i> May include aggregates from lower nesting levels
code	String	AC or LC
serial	String	Serial number included in the code, or series and number of the form (label)
inner	Array of Objects	Composition of the nested aggregate
Response body example (fragment)		
<pre>[   {     "code": "02048102230292612136813221214371200422112",     "serial": null,     "inner": [       {         "code": "0104810223029261212P99Gki6\u001d93z&amp;c:",         "serial": "DAA001052237"       },       {         "code": "0104810223029261212P99GkiS\u001d93BUtr",         "serial": "DAA001052238"       },       ...     ]   } ]</pre>		

```
        "code": "0104810223029261212P99GoAC\u001d93P(c*",
        "serial": "DAA001053436"
    }
]
},
{
    "code": "00048116129001002508",
    "serial": null,
    "inner": [
        {
            "code": "00048116129001002515",
            "serial": null,
            "inner": [
                {
                    "code": "0104810898000848212P99Wto3\u001d93C6iG",
                    "serial": "DAA001052010"
                },
                {
                    "code": "0104810898000848212P99WtMA\u001d93kd,<" ,
                    "serial": "DAA001052011"
                },
                {
                    "code": "0104810898000848212P99WtMS\u001d93jvxK",
                    "serial": "DAA001052012"
                },
                ...
            ]
        },
        {
            "code": "00048116129001002522",
            "serial": null,
            "inner": [
                {
                    "code": "0104810268023880212P99St2K\u001d93hM+W",
                    "serial": "0104810268023880212P99St2K"
                },
                {
                    "code": "0104810268023880212P99StFh\u001d93rTGY",
                    "serial": "0104810268023880212P99StFh"
                },
                {
                    "code": "0104810268023880212P99StET\u001d93u_4a",
                    "serial": "0104810268023880212P99StET"
                },
                ...
            ]
        },
        ...
    ]
},
...
{
    "code": "f5366ad6-d136-11ed-afa1-0242ac120002",
    "serial": null
}
]
```

### 3.4.7.2. Representation B

Request type: POST  
Request string: /v3/get\_aggregate\_inside  
Parameters:

**Table 3.4.7.3 HTTP request parameters**

Key	Required	Data type	Description
uuid	Yes	String	AC (SSCC or custom)
Request body example			
<pre>{   "uuid": "1323010520222111\u001d4144811612900000422112" }</pre>			

**Table 3.4.7.4 HTTP response parameters, Code 200**

Key	Data type	Description
{:uuid}	Object	Element key name corresponds to the AC value; element value contains the description and composition of the aggregate, including the aggregation tree (nesting structure taking into account child ACs). Aggregate class object (see <a href="#">4.1.11</a> )
Response body example (fragment)		
<pre>{   "1323010520222111\u001d4144811612900000422112": {     "aggregates_count": 0,     "status": "Агрегат содержит коды и/или агрегаты",     "level": 1,     "high_aggregates": [       {         "uuid": "00048116129010631669",         "level": 2       }     ],     "labels_count": 5,     "user_info": "C000 \"Свиток\"",     "aggregate_inside": {},     "labels_inside": [       "0104812845008808215CjBmJL,pugAq\u001d91FFD0\u001d92dGVzdBSSkDJP1g/BAAbCv3wv/OaXx8Fb2u5JT9pZgKc=",       "0104812845008808215&amp;)'WXeauGKm\u001d91FFD0\u001d92dGVzdPBTDspuGymZRALebENUimTf271ALnTtHAC/ZFY=",       ...       "0104812845008808215j.FUYPHZM_nV\u001d91FFD0\u001d92dGVzdBs1wK4e8qW5oRXvEE8W49YCi+uoC4R3REf5730="     ]   } }</pre>		

### 3.5 Report handling methods

#### 3.5.1 Method "Labeling report"

Request type: POST  
Request string: /v2/reports/addMark  
Parameters:

**Table 3.5.1.1** HTTP request parameters

Key	Required	Data type	Description
group	Yes	String	Product group (the list of values is given in Table 4.2.1.2).
labels	Yes	Array of Strings	Array (list) of LCs applied to the item. <b>Limits:</b> The codes listed in the array must meet the following conditions: 1. LCs must belong to a single GTIN; 2. LCs must be of the same type LabelType (see 4.2.2); 3. Maximum 25,000 LCs per request
params	No	Array of Objects	Array of additional parameters describing the batch of labeled items. The set of parameters (additional characteristics) and conditions of their setting is determined by the belonging of the labeled item to the corresponding section of the catalog and can be retrieved by the /catalogs/{:code} method (see 3.2.2). Param class objects (see 4.1.5) listed in the mark_params attribute of the Category class object (see 4.1.3 and Table D.1 of Annex D)

#### Request body example

```
{
  "group": "milk",
  "labels": [
    "0104811258000850212apa5tyq\u001d93_dwG",
    "0104811258000850212pxq5tkk\u001d93NQw^"
  ],
  "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 3.5.1.2** HTTP response parameters, Code 200

Key	Data type	Description
report_id	String	UID, a unique identifier of the labeling report (report_type = 1)
Response body example		
<pre> {   "report_id": "22331fa0-740c-4307-80a7-10b24827d9ff" } </pre>		

### 3.5.2 Method "Report on group labeling with aggregation"

Request type: POST  
 Request string: /v2/reports/addGroupMark  
 Parameters:

**Table 3.5.2.1** HTTP request parameters

Key	Required	Data type	Description
group	Yes	String	Product group (the list of values is given in Table 4.2.1.2)
packages	Yes	Array	Array of packages to be labeled and aggregated
aggregate	Yes	String	AC (SSCC or custom) applied to the package. <b>Limit:</b> Maximum 100 ACs per request
labels	Yes	Array of Strings	Array (list) of LCs applied to items in this package. <b>Limits:</b> The codes listed in the array must meet the following conditions: 1. LCs must belong to a single GTIN; 2. LCs must be of the same type LabelType (see 4.2.2); 3. Maximum 25,000 LCs per aggregate
params	No	Array of Objects	Array of additional parameters describing the batch of labeled items. The set of parameters (additional characteristics) and conditions of their setting is determined by the belonging of the labeled item to the

		<p>corresponding section of the catalog and can be retrieved by the /catalogs/{:code} method (see <a href="#">3.2.2</a>). Param class objects (see <a href="#">4.1.5</a>) listed in the mark_params attribute of the Category class object (see <a href="#">4.1.3</a> and Table <a href="#">D.1</a> of Annex <a href="#">D</a>)</p>
<b>Request body example (fragment)</b>		
<pre> {   "group": "milk",   "packages": [     {       "aggregate": "010481026802491711200927100765\u001d370012\u001d210 0600",       "labels": [         "0104811258000850212apa5tyq\u001d93_dwG",         "0104811258000850212pxq5tkk\u001d93NQw^"       ]     },     {       "aggregate": "010481026802491711200927100765\u001d370012\u001d210 0601",       "labels": [         "0104811258000850212grxc2dd\u001d93F2ds",         "0104811258000850212osbBe43\u001d938dbw"       ]     },     ...   ],   "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"     }   ] } </pre>		

```

    }
  ]
}

```

**Table 3.5.2.2** HTTP response parameters, Code 200

Key	Data type	Description
packages	Object	List of labeled and aggregated packages
{:aggregate}	Object	Element key name corresponds to the AC applied to the package; element value contains the labeling report UID
report_id	String	UID, a unique identifier of the labeling report (report_type = 1)
Response body example (fragment)		
<pre> "packages": {   "010481026802491711200927100765\u001d370012\u001d2100600": {     "report_id": "58ef4625-c8bf-4cdc-8d06-cf9111a36677"   },   "010481026802491711200927100765\u001d370012\u001d2100601": {     "report_id": "2909732b-a9ca-4a4b-b3bd-3e18b575c9e1"   },   ... } </pre>		

### 3.5.3 Method "Aggregation report"

#### 3.5.3.1. Representation A

Request type: POST

Request string: /v2/reports/addAggregate

Parameters:

**Table 3.5.3.1** HTTP request parameters

Key	Required	Data type	Description
aggregate	Yes	String	AC (SSCC or custom)
labels	Yes	Array of Strings	<p>Array of lower level LCs or ACs to be aggregated.</p> <p><b>Limits:</b> The codes listed in the array must meet the following conditions:</p> <ol style="list-style-type: none"> <li>1. LCs must belong to the same product group and have the same status;</li> <li>2. LC must not be included into another aggregate;</li> <li>3. The AC must not be empty, i.e., it must contain nested codes (child or lower level LCs or ACs);</li> <li>4. AC must not be included into the top level aggregate;</li> <li>5. The LC and the AC to which it is assigned cannot be specified at the same time;</li> <li>6. Maximum 25,000 LCs or ACs per</li> </ol>

			request.
Request body example (fragment)			
<i>Option 1.</i> Aggregation of the initial level (labeling codes)			
<pre>{   "aggregate": "000481901888000000180204811309016724376\u001d1320111230987zyx",   "labels": [     "0104811309016724212FGFq4tJ\u001d93m_c9",     "0104811309016724212Ddivj15\u001d93m2dN",     ...     "0104811309016724212Seyct2y\u001d93mKCh"   ] }</pre>			
<i>Option 2.</i> Aggregation of the second and subsequent levels (nested aggregation codes)			
<pre>{   "aggregate": "00048116129000000529",   "labels": [     "000481901888000000180204811309016724376\u001d1320111230987zyx",     "000481901888000000250204811309016724376\u001d1320111230123abc",     ...     "000481901888000000320204811309016724376\u001d1320111230k1m546"   ] }</pre>			

**Table 3.5.3.2** HTTP response parameters, Code 200

Key	Data type	Description
report_id	String	UID, a unique identifier of the aggregation report (report_type = 2)
Response body example		
<pre>{   "report_id": "22331fa0-740c-4307-80a7-10b24827d9ff" }</pre>		

### 3.5.3.2. Representation B

Request type: POST  
 Request string: /v3/reports/addAggregate  
 Parameters:

**Table 3.5.3.3** HTTP request parameters

Key	Required	Data type	Description
group	No	String	Product group (the list of values is given in Table 4.2.1.2)
aggregate	Yes	String	AC (SSCC or custom)
labels	Yes/No	Array of Strings	Array of LCs to be aggregated. <b>Limits:</b> 1. Maximum 100,000 LCs per 1 query, including total for the entire aggregation tree (nesting structure including child LCs). 2. Maximum 5 nesting levels (starting from the top aggregation level). <b>Note:</b> The codes listed in the array must meet the following conditions:

			<ol style="list-style-type: none"> <li>1. LCs must belong to the same product group and have the same status;</li> <li>2. LC must not be included into another aggregate;</li> <li>3. The LC and the AC to which it is assigned cannot be specified at the same time.</li> </ol>
aggregates	Yes/No	Array of Strings	<p>Array of lower level ACs to be aggregated. <b>Note:</b> The codes listed in the array must meet the following conditions:</p> <ol style="list-style-type: none"> <li>1. The AC must not be empty, i.e., it must contain nested codes (child or lower nesting level LCs or ACs);</li> <li>2. ACs (including nested ones) must contain LCs belonging to the same product group and having the same status;</li> <li>3. AC must not be included into the top or upper level aggregate;</li> </ol>
<b>Request body example (fragment)</b>			
<b>Option 1.</b> Aggregation of the initial level (labeling codes)			
<pre>{   "aggregate": "1323010520222111\u001d4144811612900000422112",   "labels": [     "0104812845008808215CjBmJL , pugAq\u001d91FFD0\u001d92dGVzdBSSkDJP1g/BAA bCv3wv/OaXx8Fb2u5JT9pZgKc=",     "0104812845008808215&amp;' )WXeauGKm\u001d91FFD0\u001d92dGVzdPBTdSpuGymZRA LebENUimTf271ALnTtHAC/ZFY=",     ...     "0104812845008808215j.FUYPHZM_nV\u001d91FFD0\u001d92dGVzdBs1wK4e8qW5oR XvEE8W49YCi+uoC4R3REf5730="   ] }</pre>			
<b>Option 2.</b> Aggregation of the second and subsequent levels (nested aggregation codes)			
<pre>{   "aggregate": "00048116129010631669",   "aggregates": [     "000481901888000000320204811309016724376\u001d1320111110abcdef",     "000481901888000000180204811309016724376\u001d1320111230987zyx",     ...     "000481901888000000560204811309016724376\u001d1320111230987d1e"   ] }</pre>			

**Table 3.5.3.4** HTTP response parameters, Code 200

Key	Data type	Description
report_id	Integer	Report ID
report_uid	String	UID, a unique identifier of the aggregation report (report_type = 2)
<b>Response body example</b>		
<pre>{   "report_id": 5214,   "report_uid": "cd31ab3a-f4ed-42c4-b464-95e435f3a1ca" }</pre>		

### 3.5.4 Method "UCM usage"

Request type: POST  
Request string: /v2/reports/addUsed  
Parameters:

**Table 3.5.4.1 HTTP request parameters**

Key	Required	Data type	Description
group	Yes	String	For items labeled with UCM, the value "ukz" shall be set
item	Yes	String	Item code from the catalog. Parameter code from the items list items retrieved by /catalogs/{:code} method (see 3.2.2), where code = 10090, which corresponds to the product group ukz (see Table 4.1.4.2)
labels	Yes	Array of Strings	Array (list) of series and numbers of used UCMs. <b>Limit:</b> maximum 25,000 labels per request.
Request body example (fragment)			
<pre>{   "group": "ukz",   "item": "100900000695408", //Кофе (ТН ВЭД ЕАЭС из 0901)   "labels": [     "AAA011050881",     "AAA011050882",     ...     "AAA011051000"   ] }</pre>			

**Table 3.5.4.2 HTTP response parameters, Code 200**

Key	Data type	Description
report_id	String	UID, a unique identifier of the UCM usage report (report_type = 4)
Response body example		
<pre>{   "report_id": "4cd57fc9-0fef-4857-988d-621e6ecbbdc" }</pre>		

### 3.5.5 Method "Report execution status"

Request type: POST  
Request string: /v2/reports  
Parameters:

**Table 3.5.5.1 HTTP request parameters**

Key	Required	Data type	Description
report_id	Yes	String	Unique identifier of the report. UID retrieved from methods /v2/reports/addMark (see 3.5.1), /v2/reports/addGroupMark (see 3.5.2),

		<p>/v2/reports/addAggregate (see <a href="#">3.5.3.1</a>),  /v2/reports/addUsed (see <a href="#">3.5.4</a>),  /v2/reports/addCancelUsed (see <a href="#">3.5.11</a>),  /v2/reports/addMarkCirculation (see <a href="#">3.5.9</a>),  /v2/reports/addMarkLeftovers (see <a href="#">3.5.10</a>),  /v2/reports/addMarkTrans (see <a href="#">3.5.12</a>),  /v2/reports/addManufacture (see <a href="#">3.5.14.1</a>),  /v2/reports/addChangeOwnerMade (see <a href="#">3.5.15.1</a>)  and /v2/reports/edit (see <a href="#">3.5.13</a>), as well as  /v2/reports/list (see <a href="#">3.5.6.1</a>)</p>
<b>Request body example</b>		
<pre>{   "report_id": "22331fa0-740c-4307-80a7-10b24827d9ff" }</pre>		

**Table 3.5.5.2** HTTP response parameters, Code 200

Key	Data type	Description
created_at	Date	Report creation date and time
updated_at	Date	Date and time of the last update of information or report execution
type	Integer	Report type ID (the list of values is given in <a href="#">Table 4.2.8.2</a> )
report_id	String	Unique identifier of the report (UID)
group	String	Product group (the list of values is given in <a href="#">Table 4.2.1.2</a> ). <i>Note:</i> Available only for reports generated by methods having the input parameter "group".
status	Object	Current report status. ReportStatus class object (see <a href="#">4.2.5</a> )
document_id	String	Unique identifier of the supply document. <i>Note:</i> Output for circulation reports generated by method /v2/reports/addMarkCirculation if the specified parameter has been set (see <a href="#">3.5.9</a> )
document_type	Object	Type of underlying document. Enumeration value shippingDocs obtained by /directories/{:code} method (see <a href="#">3.8.3</a> ; the list of values is given in <a href="#">Table 4.3.3</a> ). <i>Note:</i> Output for owner change reports generated by method /v2/reports/addChangeOwnerMade (see <a href="#">3.5.15.1</a> )
document_date	Date	Date of underlying document. <i>Format:</i> YYYY-MM-DD. <i>Note:</i> Output for owner change reports generated by method /v2/reports/addChangeOwnerMade (see <a href="#">3.5.15.1</a> )
user_from	Object	Information on the predecessor agent

		(former owner) that previously owned unused LCs, UCMs, PCs and/or PPDCs. Agent class object (see <a href="#">4.1.2</a> ). <i>Note:</i> Output for owner change reports (in case of reorganization of a business entity) generated by method <code>/v2/reports/addChangeOwnerMade</code> (see <a href="#">3.5.15.1</a> )
result	Object	Report processing result presentation. <i>Note:</i> Attribute set depends on the report type and the results of its execution. It may contain additional information.
labels	Object	LC processing results
aggregate	Object	AC processing results
type_id	Integer	Code type. Code ID retrieved from <code>/labels/types</code> method (see <a href="#">3.4.3</a> ; the list of values is given in Table <a href="#">4.2.2.2</a> ).

**Response body example**

**Option 1.** Report created but not yet processed

```
{
  "created_at": "2020-12-15 09:55:46+03",
  "updated_at": null,
  "type": 1,
  "report_id": "22331fa0-740c-4307-80a7-10b24827d9ff",
  "group": "shoes",
  "status": {
    "code": 0,
    "message": "Создан"
  },
  "result": null
}
```

**Option 2.** The report executed completely 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",
  "group": "tires",
  "status": {
    "code": 30,
    "message": "Выполнен"
  },
  "result": {
    "labels": {
      "success": 5
    },
    "type_id": 20
  }
}
```

**Option 3.** Labeling report with aggregation executed completely and has no 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
  },
  "type_id": 12
}
}
```

**Option 4.** Labeling and aggregation report executed partially  
(labeling done, 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",
    "type_id": 12
  }
}
```

**Option 5.** Labeling report with aggregation executed partially  
(aggregation executed completely, except for rejected labeling 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": [
        "0104810263033525212sFV261B\u001d935GVA not found",
        "0104810263033525212D2v4Ddv\u001d93dF2E not found"
      ],
      "success": 3
    },
    "aggregate": {
```

<pre>                 "success": 4             },             "type_id": 12         }     } </pre>
<b>Option 6. Aggregation report executed partially (part of labeling codes not found)</b>
<pre> {   "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": [         "01048141300000542121vtxTq2qgpL5\u001d910005\u001d92151EbB8T3A 1gtdPY72JAK9f3ouksrDcgzPtZojqs2YrJ not found",         "010481413000005421232VxgAgX1cLi\u001d910005\u001d921dwpGPBzHa 4J9DDLRTUx9RhziGVuaY2ticE3di1TVL4t not found",         ...         "0104814130000054212Y33i7WyHbhe3\u001d910005\u001d922MGuuEi3cK rAnzXmRJ6n2vpHXBgXoVY7EHXDxxR8CrrH not found"       ]     },     "aggregate": {       "success": 90     }   } } </pre>
<b>Option 7. Report not executed due to an error</b>
<pre> {   "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": "Ошибка"   },   "result": {     "error": 2001,     "message": "GTIN not found"   } } </pre>
<b>Option 8. Residual stock labeling report executed partially (one of the labeling codes was not found)</b>
<pre> {   "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", </pre>

```

    "group": "leftovers_shoes",
    "status": {
      "code": 30,
      "message": "Выполнен"
    },
    "result": {
      "labels": {
        "failed": 1,
        "details": [
          {
            "label": "010481270000039212FcDtbAMjwCfY\u001d91TEST\u001d
92testtest...test test not found",
            "comment": "Новая обувь, партия 127"
          },
          {
            "label": "0104810978916892212Y6MayiL1LqLY\u001d91TEST\u001d
92testtest...testtest",
            "comment": "Обувь женская, для вывоза из РБ"
          },
          {
            "label": "0104814130000030212srxv8YiZ4bpN\u001d91TEST\u001d
92testtest...testtest",
            "comment": "Рспродажа в Беларуси"
          },
          ...
        ],
        "success": 24
      },
      "type_id": 20
    },
    "gtin_count": {
      "04810978916892": 1,
      "04814130000030": 1,
      ...
    }
  }
}

```

**Option 9.** Circulation report executed completely and has no 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
    },
    "type_id": 7,
    "gtin_count": {
      "04811595000018": 3,
      "08808563305196": 3
    }
  }
}

```

<b>Option 10.</b> Circulation report executed partially (authenticity and correct status of a part of labeling codes was not confirmed due to various reasons)
<pre>{   "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": "0104640043462197215JDMqbLtKlYIN\u001d91FFD0\u001d 92dGVzdN3w8G+M+YxgA5vsz98pvQzh078mehnUQ7ByRkA= not found. Не подтверждено оператором. В обороте."         },         ...         {           "label": "0104640043462197152cpfgcq5pdx\u001d918097\u001d 92co51XTcPbDgmUbdy1DJMFOR1VbVUGYx/J009x1iePe3hiFxvgXfI2SHS3ycDJiZCwID1LA7zGR8cG g1NpdtBDA== not found. Не подтверждено оператором. Неверный формат."         }       ],       "success": 95     },     "gtin_count": {       "04640043462197": 95     }   } }</pre>
<b>Option 11.</b> Circulation report was executed based on a supply document from the Russian Federation, labeling codes were successfully processed
<pre>{   "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     }   } }</pre>

<b>Option 12.</b> Report on changes to the labeling parameters of a previously submitted labeling report
<pre>{   "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     }   } }</pre>
<b>Option 13.</b> Owner change report (in case of reorganization of a business entity) executed in full and without errors
<pre>{   "created_at": "2024-10-09 12:45:00+03",   "updated_at": "2024-10-09 12:45:00.22853+03",   "type": 15,   "report_id": "7ce6e142-7786-48f9-b99c-cbb25446ad73",   "document_type": {     "code": "aktreorg",     "name": "Акт передачи",     "comment": null   },   "document_date": "2024-09-30",   "user_from": {     "gln": "4810000000007",     "unp": "100412881",     "name": "Индивидуальный предприниматель Иванов И.И.",     "address": "г. Минск, ул. Красивая, д. 1, кв. 1",     "country": "112",     "is_rules_agree": true   },   "status": {     "code": 30,     "message": "Выполнен"   },   "result": {     "labels": {       "success": 56     },     "type_id": 21   } }</pre>
<b>Option 14.</b> Owner change report not executed due to errors
<pre>{   "created_at": "2024-10-04 16:50:19+03",   "updated_at": "2024-10-04 16:50:50.454781+03",   "type": 15,   "report_id": "e91522d1-dd89-4888-b89d-15edf386a034",</pre>

```

    "status": {
      "code": 40,
      "message": "Ошибка"
    },
    "result": {
      "labels": {
        "failed": 15,
        "details": [
          "0104810347004960212PvLoL3w4P2x4\u001d910005\u001d92FQLyLPCNtW
          efoagGWAZZTb9EoaQwe1fa9KSRhUkzLS2jMy6ReYUg5mAjp55hMfgNxEmbPXm7M8d6kp4LSAC4CwJt
          не найден или имеет некорректный статус/владельца.",
          "0104810347004960212LFmp9nxyZzxy\u001d910005\u001d92t7b3vshWa
          t8FmviDzRXQjXQ9NYgx8kaCCtEE2jK8dfYvMpCUjtpubq26VoMsG9WmgY2bKn9PaSsTnoJHwejqXKL
          не найден или имеет некорректный статус/владельца.",
          ...
          "AAA012345289 не найден или имеет некорректный
          статус/владельца.",
          "AAA012345290 не найден или имеет некорректный
          статус/владельца."
        ]
      }
    }
  }
}

```

### 3.5.6 Method "Report list"

#### 3.5.6.1. Representation A

Request type: POST  
Request string: /v2/reports/list  
Parameters:

**Table 3.5.6.1 HTTP request parameters**

Key	Required	Data type	Description
report_type	Yes	Integer	Report type. Report type ID retrieved from /v2/reports/types method (see 3.5.7; the list of values is given in Table 4.2.8.2).
document_id	No	String	Unique identifier of the report or unique identifier of the supply document (for circulation reports generated by method /v2/reports/addMarkCirculation if the specified parameter was set, see 3.5.9)
reason	No	String	Owner change reason. Enumeration value code change_of_owner obtained by /directories/{:code} method (see 3.8.3; the list of values is given in Table 4.3.7). <i>Note:</i> Only available for owner change reports (report_type = 15).
Request body example			
<pre> {   "report_type": 1 } </pre>			

**Table 3.5.6.2** HTTP response parameters, Code 200

Key	Data type	Description
report_list	Array of Objects	Array with a list of report states. <b>Note:</b> A maximum of 100 of the recent reports are returned
created_at	Date	Report creation date and time
updated_at	Date	Date and time of the last update of information or report execution
type	Integer	Report type ID
report_id	String	Unique identifier of the report (UID)
group	String	Product group (the list of values is given in Table 4.2.1.2). <b>Note:</b> Available only for reports generated by methods having the input parameter "group".
document_id	String	Unique identifier of the supply document. <b>Note:</b> Output for circulation reports generated by method <code>/v2/reports/addMarkCirculation</code> if the specified parameter has been set (see 3.5.9)
document_type	Object	Type of underlying document. Enumeration value <code>shippingDocs</code> obtained by <code>/directories/{:code}</code> method (see 3.8.3; the list of values is given in Table 4.3.3). <b>Note:</b> Output for owner change reports generated by method <code>/v2/reports/addChangeOwnerMade</code> (see 3.5.15.1)
document_date	Date	Date of underlying document. <b>Format:</b> YYYY-MM-DD. <b>Note:</b> Output for owner change reports generated by method <code>/v2/reports/addChangeOwnerMade</code> (see 3.5.15.1)
user_from	Object	Information on the predecessor agent (former owner) that previously owned unused LCs, UCMs, PCs and/or PPDCs. Agent class object (see 4.1.2). <b>Note:</b> Output for owner change reports (in case of reorganization of a business entity) generated by method <code>/v2/reports/addChangeOwnerMade</code> (see 3.5.15.1)
status	Object	Current report status. <code>ReportStatus</code> class object (see 4.2.5)
result	Object	Report processing result presentation. <b>Note:</b> Displayed only for report types 1, 2, 4, 6, 7, 10, 13, 14 and 15
labels	Object	LC processing results
success	Integer	Number of successful items (codes)

failed	Integer	Number of incorrect items (codes)
aggregate	Object	AC processing results
success	Integer	Number of successful items (codes)
failed	Integer	Number of incorrect items (codes)
<b>Response body example (fragment)</b>		
<b>Option 1.</b> List of labeling reports (report_type = 1)		
<pre> {   "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"       },       "result": {         "labels": {           "success": 18         },         "aggregate": {           "success": 3         }       }     },     ...     {       "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"       },       "result": {         "labels": {           "failed": 2,           "success": 7         }       }     },     ...   ] } </pre>		
<b>Option 2.</b> List of circulation reports (report_type = 7)		
<pre> {   "report_list": [     {       "created_at": "2024-09-19 15:51:23+03",       "updated_at": "2024-09-19 15:51:24.018229+03",       "type": 7, </pre>		

```

        "report_id": "2939ef5b-0f17-49dc-8171-6de072d351ff",
        "group": "shoes",
        "status": {
            "code": 30,
            "message": "Выполнен"
        },
        "result": {
            "labels": {
                "success": 2
            }
        }
    },
    ...
    {
        "created_at": "2024-09-19 15:47:32+03",
        "updated_at": "2024-09-19 15:47:32.552699+03",
        "type": 7,
        "report_id": "1f591a5d-1f5b-4cc2-919c-b2719be0ac2c",
        "group": "clothes",
        "document_id": "RU-U ECB0000000000000000000077-0002",
        "status": {
            "code": 30,
            "message": "Выполнен"
        },
        "result": {
            "labels": {
                "failed": 1
            }
        }
    },
    ...
]
}

```

**Option 3.** List of owner change reports (in case of reorganization of a business entity)  
(report\_type = 15, reason = reorganization)

```

{
    "report_list": [
        {
            "created_at": "2024-10-09 12:45:00+03",
            "updated_at": "2024-10-09 12:45:00.22853+03",
            "type": 15,
            "report_id": "7ce6e142-7786-48f9-b99c-cbb25446ad73",
            "document_type": {
                "code": "aktreorg",
                "name": "Акт передачи",
                "comment": null
            },
            "document_date": "2024-09-30",
            "user_from": {
                "gln": "4810000000007",
                "unp": "100412881",
                "name": "Индивидуальный предприниматель Иванов И.И.",
                "address": "г. Минск, ул. Красивая, д. 1, кв. 1",
                "country": "112",
                "is_rules_agree": true
            },
            "status": {

```

```

        "code": 30,
        "message": "Выполнен"
    },
    "result": {
        "labels": {
            "success": 59
        }
    }
},
...
{
    "created_at": "2024-10-04 16:50:19+03",
    "updated_at": "2024-10-04 16:50:50.454781+03",
    "type": 15,
    "report_id": "e91522d1-dd89-4888-b89d-15edf386a034",
    "status": {
        "code": 40,
        "message": "Ошибка"
    }
},
...
]
}

```

### 3.5.6.2. Representation B

Request type: GET  
 Request string: /v3/reports/list?[report\_type={:report\_type}]  
 &[page={:page}]  
 &[page\_size={:page\_size}]  
 &[ordering={:ordering}]  
 &[start\_date={:start\_date}]  
 &[end\_date={:end\_date}]  
 &[reason={:reason}]  
 &[start\_transaction\_date={:start\_transaction\_date}]  
 &[end\_transaction\_date={:end\_transaction\_date}]  
 &[start\_document\_date={:start\_document\_date}]  
 &[end\_document\_date={:end\_document\_date}]  
 &[document\_number={:document\_number}]  
 &[group={:group}]  
 &[search={:search}]

Parameters:

**Table 3.5.6.3** HTTP request parameters

Key	Required	Data type	Description
report_type	No	Integer	Report type. Report type ID retrieved from /v2/reports/types method (see <a href="#">3.5.7</a> ; the list of values is given in <a href="#">Table 4.2.8.2</a> ).
page	No	Integer	Report list page number

page_size	No	Integer	Number of entries per page. <b>Note:</b> Maximum 200, by default, up to 15 per page.
ordering	No	String	Data sorting conditions: created_at: by creation date and time in ascending order; -created_at: by creation date and time in descending order
start_date	No	Date	Report creation start date. <b>Format:</b> YYYY-MM-DD
end_date	No	Date	Report creation end date. <b>Format:</b> YYYY-MM-DD
reason	No	String	Depends on the report type: 1. For label discard reports (report_type = 11): discard reason: the code of the reasons enumeration value obtained by the /directories/{:code} method (see 3.8.3; the list of values is given in Table 4.3.4). 2. For owner change reports (report_type = 15): owner change reason; the code of the change_of_owner enumeration value obtained by the /directories/{:code} method (see 3.8.3; the list of values is given in Table 4.3.7).
start_transaction_date	No	Date	Operation start date. <b>Format:</b> YYYY-MM-DD. <b>Note:</b> Used for label discard reports (report_type = 11)
end_transaction_date	No	Date	Operation end date. <b>Format:</b> YYYY-MM-DD. <b>Note:</b> Used for label discard reports (report_type = 11)
start_document_date	No	Date	Start date of underlying document. <b>Format:</b> YYYY-MM-DD. <b>Note:</b> Used for label discard reports (report_type = 11) or owner change reports (report_type = 15)
end_document_date	No	Date	End date of underlying document. <b>Format:</b> YYYY-MM-DD. <b>Note:</b> Used for label discard reports (report_type = 11) or owner change reports (report_type = 15)
document_number	No	String	Number of underlying document. <b>Note:</b> Used for label discard reports (report_type = 11) or owner change reports (report_type = 15)
group	No	String	Product group (the list of values is given in Table 4.2.1.2)

search	No	String	UID, unique identifier of the report searched in the list
<b>Request string example</b>			
<b>Option 1.</b> List of aggregation reports (report_type = 2)			
/v3/reports/list?report_type=2			
<b>Option 2.</b> List of label discard reports (report_type = 11)			
/v3/reports/list?report_type=11&page=1&page_size=15&ordering=-created_at			
<b>Option 3.</b> List of manufacturing parameters with selection by product group (report_type = 14)			
/v3/reports/list?report_type=14&page=1&page_size=15&ordering=-created_at&group=dietary_supplements			
<b>Option 4.</b> Search of a manufacturing parameter report by UID (report_type = 14)			
/v3/reports/list?report_type=14&search=fcacc009-a2ac-49b8-9271-71669eefec21			
<b>Option 5.</b> List of owner change reports (in case of reorganization of a business entity) with selection by creation date (report_type = 15, reason = reorganization)			
/v3/reports/list?report_type=15&page=1&page_size=15&ordering=-created_at&start_date=2025-01-03&end_date=2025-01-22&reason=reorganization			
<b>Option 6.</b> List of labeled goods owner change reports (report_type = 15, reason = change_owner_goods)			
/v3/reports/list?report_type=15&page=1&page_size=15&ordering=-created_at&reason=change_owner_goods			
<b>Option 7.</b> List of reports on deletion of codes from aggregate (report_type = 16)			
/v3/reports/list?report_type=16&page=1&page_size=50			

**Table 3.5.6.4 HTTP response parameters, Code 200**

Key	Data type	Description
next	String	Link to the next page of the report list. <i>Note:</i> Null if none.
previous	String	Link to the previous page of the report list. <i>Note:</i> Null if none.
results	Array of Objects	Array with the list of reports and their processing results. <i>Note:</i> Attribute set depends on the report type and the results of its execution. It may contain additional information.
id	Integer	Report ID
uid	String	Unique identifier of the report (UID)
created_at	Date	Report creation date and time
updated_at	Date	Date and time of the last update of information or report execution
type	Integer	Report type ID
labels_count	Object	Number of processed codes:
all	Integer	- total;
success	Integer	- successful;
failed	Integer	- incorrect
report_reason	String	Information message
types	Array of	Array of code type IDs used in report

	Integer	generation and processing (the list of values is given in Table <a href="#">4.2.2.2</a> )
aggregate_in	String	AC in which LCs or child ACs were included. <b>Note:</b> Output for aggregation reports (report_type = 2)
aggregated	Integer	Total number of codes added to aggregate. <b>Note:</b> Output for aggregation reports (report_type = 2)
deleted_from_aggregate	Integer	Number of codes deleted from aggregation. <b>Note:</b> Output for reports on deletion of codes from aggregate (report_type = 16)
files	Object	List of LC file names used in report generation and processing
{:file_type}	String	Element key name corresponds to the type of information returned: labels_good: successfully processed LCs; labels_error: incorrect LCs; aggregates_good: successfully processed ACs; aggregates_error: incorrect ACs; destroy_aggregates: destroyed ACs. Key value contains the file name.
group	String	Product group (the list of values is given in Table <a href="#">4.2.1.2</a> ). <b>Note:</b> Available only for reports generated by methods having the input parameter "group".
user_info	Object	Details of the circulation participant generated the report. Agent class object (see <a href="#">4.1.2</a> )
status	Object	Current report status. ReportStatus class object (see <a href="#">4.2.5</a> )
params	Array of Objects	Array of parameters describing the following based on the report type: 1. For label discard reports (report_type = 11): information on the discard reason and the underlying document. 2. For manufacturing parameter reports (report_type = 14): information on the date of manufacture (production) of the item, date of expiration and other characteristics of the unit of labeled item (if any). 3. For owner change reports (report_type = 15): information on the owner change reason and the underlying document. Param class objects (see <a href="#">4.1.5</a> )
<b>Response body example (fragment)</b>		
<b>Option 1.</b> List of aggregation reports (report_type = 2)		
<pre>{   "next": "http://api.datamark.by/v3/reports/list?page=2&amp;report_type=2",</pre>		

```
"previous": null,
"results": [
  ...
  {
    "id": 5214,
    "uid": "cd31ab3a-f4ed-42c4-b464-95e435f3a1ca",
    "created_at": "2025-01-03T12:55:07.869284",
    "updated_at": "2025-01-03T12:55:08.095751",
    "type": 2,
    "labels_count": {
      "all": 5,
      "failed": 0,
      "success": 5,
      "aggregated": 5
    },
    "types": [
      7
    ],
    "aggregate_in": "1323010520222111\u001d414481161290000422112",
    "aggregated": 5,
    "files": {
      "labels_good": "report-cd31ab3a-f4ed-42c4-b464-95e435f3a1ca-
labels_good-nIuWdp.csv",
      "labels_error": null,
      "aggregates_good": null,
      "aggregates_error": null
    },
    "user_info": {
      "user_info_id": 129,
      "name": "C000 \\"Свиток\\"",
      "address": "г. Минск, ул. Тимирязева, д.2, ком. 16",
      "unp": "190704463",
      "gln": "4811612900000",
      "country": {
        "code": "112",
        "name": "Беларусь"
      }
    },
    "status": {
      "code": 30,
      "message": "Выполнен"
    }
  },
  ...
]
```

**Option 2.** List of label discard reports  
(report\_type = 11)

```
{
  "next": null,
  "previous": null,
  "results": [
    {
      "id": 5443,
      "uid": "0d3e86d3-ddf0-4664-a612-4fa9130bd7df",
      "created_at": "2025-01-12T18:21:41.006824",
      "updated_at": "2025-01-12T18:21:41.253681",
```

```
"type": 11,
"labels_count": {
  "all": 5,
  "failed": 0,
  "success": 5
},
"types": [
  7
],
"params": {
  "transaction_date": {
    "type": "date",
    "value": "2025-01-10",
    "display_name": "Дата операции",
    "display_value": "10.01.2025"
  },
  "document_type": {
    "type": "directory",
    "value": "akt",
    "display_name": "Тип документа",
    "display_value": "АКТ"
  },
  "document_date": {
    "type": "date",
    "value": "2025-01-12",
    "display_name": "Дата документа",
    "display_value": "12.01.2025"
  },
  "document_number": {
    "type": "string",
    "value": "2a",
    "display_name": "Номер документа",
    "display_value": "1"
  },
  "reason": {
    "type": "directory",
    "value": "damage",
    "display_name": "Причина",
    "display_value": "Порча"
  }
},
"files": {
  "labels_good": "report-0d3e86d3-ddf0-4664-a612-4fa9130bd7df-
labels_good-wSyvjD.csv",
  "destroy_aggregates": "report-0d3e86d3-ddf0-4664-a612-
4fa9130bd7df-destroy_aggregates-sANWXA.csv"
},
"user_info": {
  "user_info_id": 129,
  "name": "C000 \"Свиток\"",
  "address": "г. Минск, ул. Тимирязева, д.2, ком. 16",
  "unp": "190704463",
  "gln": "4811612900000",
  "country": {
    "code": "112",
    "name": "Беларусь"
  }
}
```

```
    },  
    "status": {  
      "code": 30,  
      "message": "Выполнен"  
    }  
  },  
  ...  
]  
}
```

**Option 3.** List of manufacturing parameters with selection by product group  
(report\_type = 14)

```
{  
  "next": "http://api.datamark.by/v3/reports/list?ordering=-  
created_at&page=2&report_type=14",  
  "previous": null,  
  "results": [  
    {  
      "id": 3685,  
      "uid": "bd1bb5e7-64cd-4d30-92b9-82914db648f6",  
      "user_info_to": null,  
      "created_at": "2024-11-30T21:36:07.017991",  
      "updated_at": "2024-11-30T21:36:07.427326",  
      "type": 14,  
      "labels_count": {  
        "all": 12,  
        "failed": 0,  
        "success": 12  
      },  
      "types": [  
        7  
      ],  
      "gtin": "",  
      "files": {  
        "labels_good": "report-bd1bb5e7-64cd-4d30-92b9-82914db648f6-  
labels_good-nhyYTf.csv"  
      },  
      "group": {  
        "code": "dietary_supplements",  
        "name": "Биологически активные добавки к пище"  
      },  
      "user_info": {  
        "user_info_id": 129,  
        "name": "С000 \Свиток\\"",  
        "address": "г. Минск, ул. Тимирязева, д.2, ком. 16",  
        "unp": "190704463",  
        "gln": "4811612900000",  
        "country": {  
          "code": "112",  
          "name": "Беларусь"  
        }  
      }  
    },  
    "status": {  
      "code": 30,  
      "message": "Выполнен"  
    },  
    "params": [  
      {
```

```
        "slug": "manufacture_date",
        "value": "2024-11-30",
        "display_value": "30.11.2024",
        "display_name": "Дата изготовления (производства) товара"
    },
    {
        "slug": "expiration_date",
        "value": "2025-03-16",
        "display_value": "16.03.2025",
        "display_name": "Дата истечения срока годности"
    }
]
},
{
    "id": 3684,
    "uid": "fcacc009-a2ac-49b8-9271-71669eefec21",
    "user_info_to": null,
    "created_at": "2024-11-30T21:33:56.113449",
    "updated_at": "2024-11-30T21:33:56.666390",
    "type": 14,
    "labels_count": {
        "all": 5,
        "failed": 1,
        "success": 4
    },
    "report_reason": "В отчете есть ошибочные коды",
    "types": [
        7
    ],
    "gtin": "",
    "files": {
        "labels_good": "report-fcacc009-a2ac-49b8-9271-71669eefec21-
labels_good-zufIMG.csv",
        "labels_error": "report-fcacc009-a2ac-49b8-9271-71669eefec21-
labels_error-bxyhMU.csv"
    },
    "group": {
        "code": "dietary_supplements",
        "name": "Биологически активные добавки к пище"
    },
    "user_info": {
        "user_info_id": 129,
        "name": "C000 \"Свиток\"",
        "address": "г. Минск, ул. Тимирязева, д.2, ком. 16",
        "unp": "190704463",
        "gln": "4811612900000",
        "country": {
            "code": "112",
            "name": "Беларусь"
        }
    },
    "status": {
        "code": 25,
        "message": "Выполнен частично"
    },
    "params": [
        {
```

```

        "slug": "manufacture_date",
        "value": "2024-11-29",
        "display_value": "29.11.2024",
        "display_name": "Дата изготовления (производства) товара"
    },
    {
        "slug": "expiration_date",
        "value": "2026-11-28",
        "display_value": "28.11.2026",
        "display_name": "Дата истечения срока годности"
    }
]
},
...
]
}

```

**Option 4.** Search of a manufacturing parameter report by UID  
(report\_type = 14)

```

{
  "next": null,
  "previous": null,
  "results": [
    {
      "id": 3684,
      "uid": "fcacc009-a2ac-49b8-9271-71669eefec21",
      "user_info_to": null,
      "created_at": "2024-11-30T21:33:56.113449",
      "updated_at": "2024-11-30T21:33:56.666390",
      "type": 14,
      "labels_count": {
        "all": 5,
        "failed": 1,
        "success": 4
      },
      "report_reason": "В отчете есть ошибочные коды",
      "types": [
        7
      ],
      "gtin": "",
      "files": {
        "labels_good": "report-fcacc009-a2ac-49b8-9271-71669eefec21-labels_good-zufIMG.csv",
        "labels_error": "report-fcacc009-a2ac-49b8-9271-71669eefec21-labels_error-bxyhMU.csv"
      },
      "group": {
        "code": "dietary_supplements",
        "name": "Биологически активные добавки к пище"
      },
      "user_info": {
        "user_info_id": 129,
        "name": "C000 \Свиток\\"",
        "address": "г. Минск, ул. Тимирязева, д.2, ком. 16",
        "unp": "190704463",
        "gln": "4811612900000",
        "country": {
          "code": "112",

```

```
        "name": "Беларусь"
      }
    },
    "status": {
      "code": 25,
      "message": "Выполнен частично"
    },
    "params": [
      {
        "slug": "manufacture_date",
        "value": "2024-11-29",
        "display_value": "29.11.2024",
        "display_name": "Дата изготовления (производства) товара"
      },
      {
        "slug": "expiration_date",
        "value": "2026-11-28",
        "display_value": "28.11.2026",
        "display_name": "Дата истечения срока годности"
      }
    ]
  }
]
```

**Option 5.** List of owner change reports (in case of reorganization of a business entity) with selection by creation date (`report_type = 15, reason = reorganization`)

```
{
  "next": null,
  "previous": null,
  "results": [
    {
      "id": 5321,
      "uid": "c364a0ac-9458-41d7-9aed-8c63d9b957af",
      "created_at": "2025-01-10T10:13:35.918249",
      "updated_at": "2025-01-10T10:13:36.112184",
      "type": 15,
      "labels_count": {
        "all": 10,
        "failed": 0,
        "success": 10
      },
      "types": [
        7
      ],
      "params": {
        "document_type": {
          "type": "directory",
          "value": "aktreorg",
          "display_name": "Тип документа",
          "display_value": "Акт передачи"
        },
        "document_date": {
          "type": "date",
          "value": "2025-01-10",
          "display_name": "Дата документа",
          "display_value": "10.01.2025"
        }
      }
    }
  ]
}
```

```
        "document_number": {
            "type": "string",
            "value": "1001/1",
            "display_name": "Номер документа",
            "display_value": "1001/1"
        },
        "reason": {
            "type": "directory",
            "value": "reorganization",
            "display_name": "Причина",
            "display_value": "Реорганизация"
        },
        "comment": {
            "type": "string",
            "value": "Переход ИП в созданное им ЮЛ",
            "display_name": "Комментарий",
            "display_value": "Переход ИП в созданное им ЮЛ"
        }
    },
    "files": {
        "labels_good": "report-c364a0ac-9458-41d7-9aed-8c63d9b957af-
labels_good-NfZTSf.csv",
        "labels_error": null,
        "aggregates_good": null
    },
    "group": {
        "code": "dietary_supplements",
        "name": "Биологически активные добавки к пище"
    },
    "user_info": {
        "user_info_id": 129,
        "name": "C000 \"Свиток\"",
        "address": "г. Минск, ул. Тимирязева, д.2, ком. 16",
        "unp": "190704463",
        "gln": "4811612900000",
        "country": {
            "code": "112",
            "name": "Беларусь"
        }
    },
    "user_info_from": {
        "user_info_id": 10564,
        "name": "Индивидуальный предприниматель Иванов И.И.",
        "address": "г. Минск, ул. Красивая, д. 1, кв. 1",
        "unp": "100412881",
        "gln": "4810000000007",
        "country": {
            "code": "112",
            "name": "Беларусь"
        }
    },
    "status": {
        "code": 30,
        "message": "Выполнен"
    }
}
]
```

<b>Option 6.</b> List of labeled goods owner change reports (report_type = 15, reason = change_owner_goods)
<pre>{   "next": null,   "previous": null,   "results": [     {       "id": 5331,       "uid": "02b62b6c-2d32-4f1e-9f57-9fee9a18cf99",       "created_at": "2025-01-10T11:30:40.672400",       "updated_at": "2025-01-10T11:30:40.849588",       "type": 15,       "labels_count": {         "all": 5,         "failed": 0,         "success": 5       },       "types": [         7       ],       "params": {         "reason": {           "type": "directory",           "value": "change_owner_goods",           "display_name": "Причина",           "display_value": "Смена владельца товара"         }       },       "files": {         "labels_good": "report-02b62b6c-2d32-4f1e-9f57-9fee9a18cf99- labels_good-uqUXui.csv",         "labels_error": null,         "aggregates_good": null       },       "group": {         "code": "dietary_supplements",         "name": "Биологически активные добавки к пище"       },       "user_info": {         "user_info_id": 129,         "name": "C000 \"Свиток\"",         "address": "г. Минск, ул. Тимирязева, д.2, ком. 16",         "unp": "190704463",         "gln": "4811612900000",         "country": {           "code": "112",           "name": "Беларусь"         }       },       "status": {         "code": 30,         "message": "Выполнен"       }     },     ...   ] }</pre>

<b>Option 7.</b> List of reports on deletion of codes from aggregate processed with an error (report_type = 16)
<pre>{   "next": null,   "previous": null,   "results": [     ...     {       "id": 5459,       "uid": "eb1e7507-aa31-4021-ba8e-f6c4b4331d9a",       "created_at": "2025-01-13T13:42:31.289144",       "updated_at": "2025-01-13T13:42:31.542276",       "type": 16,       "labels_count": {         "all": 5,         "failed": 5,         "success": 0       },       "report_reason": "В отчете содержатся ошибочные КМ",       "types": [],       "aggregate_in": null,       "deleted_from_aggregate": 0,       "files": {         "labels_good": null,         "labels_error": "report-eb1e7507-aa31-4021-ba8e-f6c4b4331d9a- labels_error-lShBbC.csv",         "aggregates_good": null,         "aggregates_error": null       },       "user_info": {         "user_info_id": 129,         "name": "C000 \"Свиток\"",         "address": "г. Минск, ул. Тимирязева, д.2, ком. 16",         "unp": "190704463",         "gln": "4811612900000",         "country": {           "code": "112",           "name": "Беларусь"         }       }     },     {       "status": {         "code": 40,         "message": "Ошибка"       }     }   ],   ... }</pre>

### 3.5.7 Method "Report types"

Request type: POST  
Request string: /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 Objects	Report type list array ReportType class object (see <a href="#">4.2.8</a> )
<b>Response body example (fragment)</b>		
<pre> {   "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": 11,       "name": "Списание"     },     {       "id": 13,       "name": "Изменение параметров маркировки"     },     {       "id": 14,       "name": "Отчет о параметрах производства"     },     {       "id": 15,       "name": "Смена владельца"     },     {       "id": 16, </pre>		

```

        "name": "Удаление кодов из агрегата"
    }
]
}

```

### 3.5.8 Method "Used code list"

#### 3.5.8.1. Representation A

Request type: POST  
Request string: /v2/reports/labels  
Parameters:

**Table 3.5.8.1** HTTP request parameters

Key	Required	Data type	Description
report_id	Yes	String	Unique identifier of the report. UID retrieved from methods /v2/reports/addMark (see <a href="#">3.5.1</a> ), /v2/reports/addGroupMark (see <a href="#">3.5.2</a> ), /v2/reports/addAggregate (see <a href="#">3.5.3.1</a> ), /v2/reports/addUsed (see <a href="#">3.5.4</a> ), /v2/reports/addCancelUsed (see <a href="#">3.5.11</a> ), /v2/reports/addMarkCirculation (see <a href="#">3.5.9</a> ), /v2/reports/addMarkLeftovers (see <a href="#">3.5.10</a> ), /v2/reports/addMarkTrans (see <a href="#">3.5.12</a> ), /v2/reports/addChangeOwnerMade (see <a href="#">3.5.15.1</a> ), as well as /v2/reports/list (see <a href="#">3.5.6.1</a> )
Request body example			
<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	Unique identifier of the report (UID)
labels	Array of Strings	Array (list) of LCs, ACs or series and numbers of UCMs used in report generation
Response body example (fragment)		
<pre> {   "report_id": "a83e9a31-d025-4927-9d6f-95d7fae5d58e",   "labels": [     "0104810223007436212PWKhQkD\u001d931qxa",     "0104810223007436212PWKhQk5\u001d93RaEC",     ...     "0104810223007436212PWKhQWS\u001d932AxX"   ] } </pre>		

#### 3.5.8.2. Representation B

Request type: POST  
Request string: /v3/reports/downloads

Parameters:

**Table 3.5.8.3 HTTP request parameters**

Key	Required	Data type	Description
report_id	Yes/No	Integer	Report ID retrieved from methods /v3/labels/destroy (see <a href="#">3.4.4.2</a> ); /v3/reports/addAggregate (see <a href="#">3.5.3.2</a> ), /v3/reports/addManufacture (see <a href="#">3.5.14.2</a> ), /v3/reports/addChangeOwnerMade (see <a href="#">3.5.15.2</a> ), /v3/reports/aggregateDelete (see <a href="#">3.5.16</a> ), as well as /v3/reports/list (see <a href="#">3.5.6.2</a> ). <i>Note:</i> Must be set if report_uid is not specified
report_uid	Yes/No	String	Unique identifier of the report. UID retrieved from methods /v3/labels/destroy (see <a href="#">3.4.4.2</a> ); /v3/reports/addAggregate (see <a href="#">3.5.3.2</a> ), /v3/reports/addManufacture (see <a href="#">3.5.14.2</a> ), /v3/reports/addChangeOwnerMade (see <a href="#">3.5.15.2</a> ), /v3/reports/aggregateDelete (see <a href="#">3.5.16</a> ), as well as /v3/reports/list (see <a href="#">3.5.6.2</a> ). <i>Note:</i> Must be set if report_id is not specified
file_type	Yes	String	Type of information returned: labels_good: successfully processed LCs; labels_error: incorrect LCs; aggregates_good: successfully processed ACs; aggregates_error: incorrect ACs; destroy_aggregates: destroyed ACs
Request body example			
<i>Option 1.</i> Receiving the list of successfully processed LCs			
<pre>{   "report_uid": "fcacc009-a2ac-49b8-9271-71669eefec21",   "file_type": "labels_good" }</pre>			
<i>Option 2.</i> Receiving the list of LCs processed with an error			
<pre>{   "report_id": 3684,   "file_type": "labels_error" }</pre>			
<i>Option 3.</i> Receiving the list of successfully processed ACs			
<pre>{   "report_uid": "082945d4-dcfc-47e3-ad29-1628d6892aee",   "file_type": "aggregates_good" }</pre>			

*Note:* The request must always contain at least one of the parameters identifying the report ("report\_id" or "report\_uid") for which the information about used LCs or ACs is requested.

**Table 3.5.8.4 HTTP response parameters, Code 200**

Key	Data type	Description
labels	Array of Strings	Array (list) of LCs or ACs used in report

		generation and processing. <i>Note:</i> In case of incorrect LCs or ACs, a description of the error can be additionally specified
<b>Response body example (fragment)</b>		
<i>Option 1.</i> List of successfully processed LCs		
<pre>{   "labels": [     "0104812845008808215?&amp;T.nniB,0&amp;u\u001d91FFD0\u001d92dGVzdLRHvQ6LVjFS4H     woxLFJrFwQsXDTXbpz4bEcIRA=",     "0104812845008808215&amp;)'WXeauGKm\u001d91FFD0\u001d92dGVzdPBTdSpuGymZRA     LebENUimTf271ALnTtHAC/ZFY=",     ...     "0104812845008808215(neIbkaQptB?\u001d91FFD0\u001d92dGVzdEgtw1La2hx40B     lz73AQU2EbpVHvAvmeknf9+YM="   ] }</pre>		
<i>Option 2.</i> List of LCs processed with an error		
<pre>{   "labels": [     "0104812845008808215j.FUYPHZM_nV\u001d91FFD0\u001dd2dGVzdBs1wK4e8qw5oR     XvEE8W49YCi+uoC4R3REf5730= Код маркировки не найден",     ...   ] }</pre>		
<i>Option 3.</i> List of successfully processed ACs		
<pre>{   "labels": [     "000481901888000000180204811309016724376\u001d1320111230987zyx",     "000481901888000000250204811309016724376\u001d1320111230123abc",     ...     "000481901888000000320204811309016724376\u001d1320111230k1m546"   ] }</pre>		

### 3.5.9 Method "Circulation report"

Request type: POST  
 Request string: /v2/reports/addMarkCirculation  
 Parameters:

**Table 3.5.9.1** HTTP request parameters

Key	Required	Data type	Description
group	Yes	String	Product group (the list of values is given in Table 4.2.1.2)
labels	Yes	Array of Strings	Array (list) of LCs applied to the items. <i>Constraint:</i> Maximum 5,000 LCs per 1 request
circulation_reason	Yes	String	Reason for circulation. Enumeration value code circulation_reason obtained

			by <code>/directories/{:code}</code> method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.5</a> )
document_id	No	String	Unique identifier of the document with supply type <code>type = ship</code> or <code>type = export</code> by which the acceptance is performed, obtained by method <code>/ships/importList</code> (see <a href="#">3.7.8</a> ). <b>Note:</b> Specified if the reason is <code>circulation_reason = import</code> and the supply information is sent by the exporter and received in SIS "Electronic Mark".
params	No	Object	Additional parameters describing a batch of items. <b>Note:</b> Required if the reason is <code>circulation_reason = import</code> or <code>circulation_reason = import_third_countries</code>
<b>Option 1.</b> Import from EAEU countries ( <code>circulation_reason = import</code> )			
country	Yes	String	The numeric code of the country where the item was purchased. Enumeration value code <code>country</code> obtained by <code>/directories/{:code}</code> method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.1</a> )
document_date	Yes	Date	Date of the document confirming the purchase of the item. <b>Format:</b> YYYY-MM-DD
document_number	Yes	String	Number of the document confirming the purchase of the goods.
org_number	No	String	Taxpayer's Number or other equivalent identifier of the seller organization
org_name	No	String	Seller organization name
<b>Option 2.</b> Import from third countries ( <code>circulation_reason = import_third_countries</code> )			
country	Yes	String	The numeric code of the country where the item was purchased. Enumeration value code <code>country</code> obtained by <code>/directories/{:code}</code> method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.1</a> )
item_name	No	String	Item name

manufacture_date	Yes	Date	Manufacture (production) date. <b>Format:</b> YYYY-MM-DD
partia	No	String	Batch number
declaration_reg_number	Yes	String	Registration number of the item declaration. <b>Note:</b> The value should correspond to the established format given in Note 6 to Table <a href="#">D.1</a> of Annex <a href="#">D</a>
declaration_release_date	Yes	Date	Date of release of goods in the item declaration. <b>Format:</b> YYYY-MM-DD
declaration_ordinal_number	Yes	String	Serial number of goods in the item declaration. <b>Template:</b> [0-9]{1-3}
contract	No	String	Contract number
contract_date	No	Date	Contract date. <b>Format:</b> YYYY-MM-DD
mark_method	Yes	String	Labeling method (polygraphic protection): 1 – use of PPDCs; 2 – use of PCs; 3 – without protection.
<b>Request body example (fragment)</b>			
<pre>{   "group": "shoes",   "labels": [     "0104609432871247212JvdDBvxlfsvH\u001d91TEST\u001d92testtest...test test",     "0104609432871247212bpnSBlxsCSvw\u001d91TEST\u001d92testtest...test test",     ...     "010460943287124721eSsgGsEHcwjFs\u001d91TEST\u001d92testtest...test test"   ],   "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": "000 \u041e\u0410\u0415\u0418\u0418"   } }</pre>			

**Table 3.5.9.2** HTTP response parameters, Code 200

Key	Data type	Description
report_id	String	UID, a unique identifier of the circulation report (report_type = 7)
<b>Response body example</b>		
<pre>{   "report_id": "9f7a3581-9447-425a-a931-49412b3fdb19" }</pre>		

### 3.5.10 Method "Residual stock labeling report"

Request type: POST  
Request string: /v2/reports/addMarkLeftovers  
Parameters:

**Table 3.5.10.1** HTTP request parameters

Key	Required	Data type	Description
group	Yes	String	Product group (the list of values is given in Table 4.2.1.2). <b>Note:</b> The product group where residual stock shall be labeled is specified.
labels	Yes	Array	Array (list) of LCs applied to the residual stock. <b>Constraint:</b> Maximum 5,000 LCs per 1 request
label	Yes	String	LC
comment	No	String	Comment, note
params	Yes	Array of Objects	Array of additional parameters describing the batch of labeled residual stock. Param class objects (see 4.1.5). <b>Note:</b> The only parameter mark_method ("Method of labeling (polygraphic protection)") is used for residual stock labeling.
Request body example			
<pre>{   "group": "shoes",   "labels": [     {       "label": "0104814130000030212qc7v9aauLzpw\u001d91TEST\u001d92test       test...testtest",       "comment": "Сапоги мужские, остатки"     },     {       "label": "0104609432871247215DsgGsEHcwjFs\u001d91TEST\u001d92test       test...testtest",       "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, a unique identifier of the residual stock circulation (labeling) report (report_type = 6)

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

**Note:** The method can be applied only from the date of enabling mandatory labeling and becomes unavailable after the end of the time period for transmitting information on the application of identification means to the residual stock established for the relevant product group.

### 3.5.11 Method "UCM cancellation"

Request type: POST  
Request string: /v2/reports/addCancelUsed  
Parameters:

**Table 3.5.11.1** HTTP request parameters

Key	Required	Data type	Description
labels	Yes	Array of Strings	Array (list) of series and numbers of UCMs excluded from the usage report. <b>Limit:</b> maximum 25,000 labels per request.
Request body example (fragment)			
<pre>{   "labels": [     "AAA011050881",     "AAA011050882",     "AAA011050883",     ...     "AAA011051000"   ] }</pre>			

**Table 3.5.11.2** HTTP response parameters, Code 200

Key	Data type	Description
report_id	String	UID, a unique identifier of the UCM cancellation report (report_type = 8)
Response body example		
<pre>{   "report_id": "e15c7739-7e53-4df6-8f36-72bcf3c04b21" }</pre>		

### 3.5.12 Method "Report on labeling of goods made from customer-owned raw materials"

Request type: POST  
Request string: /v2/reports/addMarkTrans  
Parameters:

**Table 3.5.12.1** HTTP request parameters

Key	Required	Data type	Description
group	Yes	String	Product group (the list of values is given in

			Table <a href="#">4.2.1.2</a> ).
labels	Yes	Array of Strings	Array (list) of LCs applied to the item
<b>Request body example</b>			
<pre>{   "group": "milk",   "labels": [     "0104811258000850212apa5tyq\u001d93_dwG",     "0104811258000850212pxq5tkk\u001d93NQw^",     ...     "0104811258000850212fdkrDwx\u001d93dRst"   ] }</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 labeling of goods made from customer-owned raw materials (report_type = 10)
<b>Response body example</b>		
<pre>{   "report_id": "d9ba0436-70ee-449f-8c53-5a3624a32b9f" }</pre>		

### 3.5.13 Method "Labeling parameter change report"

Request type: POST  
Request string: /v2/reports/edit  
Parameters:

**Table 3.5.13.1** HTTP request parameters

Key	Required	Data type	Description
report_id	Yes	String	Unique identifier of the report to be changed. UID retrieved from methods /v2/reports/addMark (see <a href="#">3.5.1</a> ) or /v2/reports/addGroupMark (see <a href="#">3.5.2</a> ). <b>Note:</b> Only available for labeling/group labeling reports (report_type = 1).
params	Yes	Array of Objects	Array of changed parameters. The set of parameters (additional characteristics) and conditions of their setting is determined by the belonging of the labeled item to the corresponding section of the catalog and can be retrieved by the /catalogs/{:code} method (see <a href="#">3.2.2</a> ). Param class objects (see <a href="#">4.1.5</a> ) listed in the mark_params attribute of the Category class object (see <a href="#">4.1.3</a> and Table <a href="#">D.1</a> of Annex <a href="#">D</a> )
<b>Request body example</b>			
<pre>{   "report_id": "57fb2260-d978-4793-a674-27da1f7583d5",   "params": [     {</pre>			

```

    "code": 25,
    "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 3.5.13.2** HTTP response parameters, Code 200

Key	Data type	Description
report_id	String	UID, a unique identifier of the labeling parameter change report (report_type = 13)
<b>Response body example</b>		
<pre> {   "report_id": "67db7b9d-ed03-4347-a849-753eed01076c" } </pre>		

### 3.5.14 Method "Manufacturing parameter report"

#### 3.5.14.1. Representation A

Request type: POST

Request string: /v2/reports/addManufacture

Parameters:

**Table 3.5.14.1** HTTP request parameters

Key	Required	Data type	Description
group	Yes	String	Product group (the list of values is given in Table 4.2.1.2). <b>Note:</b> Available only for the product groups "Packaged water" (group = water) and "Non-alcoholic beverages and juices" (group = juice)
labels	Yes	Array of Strings	Array (list) of LCs applied to the item. <b>Limits:</b> 1. Only LCs of the Russian format (code type label_type = 7) with the status "Applied" (status = 47) or "Labeled

			items" ( <code>status = 50</code> ) are allowed. 2. Maximum 10,000 LCs per request.
<code>params</code>	Yes	Object	Manufacturing parameters
<code>manufacture_date</code>	Yes	Date	Manufacture (production) date. <b>Format:</b> YYYY-MM-DD.
<code>expiration_date</code>	Yes	Date	Expiration date. <b>Format:</b> YYYY-MM-DD.
<b>Request body example (fragment)</b>			
<pre>{   "group": "juice",   "labels": [     "0104810417004258215z2wgkvtnqjgs\u001d93dGVz",     "0104810417004258215ef1gpeqbqqyu\u001d93dGVz",     "01048104170042582154sppwmrcx272\u001d93dGVz",     ...     "0104810417004258215cnd5k2guhsec\u001d93dGVz"   ],   "params": {     "manufacture_date": "2023-12-13",     "expiration_date": "2024-04-12"   } }</pre>			

**Table 3.5.14.2** HTTP response parameters, Code 200

Key	Data type	Description
<code>report_id</code>	String	UID, a unique identifier of the manufacturing parameter report ( <code>report_type = 14</code> )
<b>Response body example</b>		
<pre>{   "report_id": "2eb185e6-ca27-401d-b21f-1363d7f0c195" }</pre>		

### 3.5.14.2. Representation B

Request type: POST  
Request string: `/v3/reports/addManufacture`  
Parameters:

**Table 3.5.14.3** HTTP request parameters

Key	Required	Data type	Description
<code>group</code>	Yes	String	Product group (the list of values is given in Table 4.2.1.2). <b>Note:</b> Available only for the product groups "Dietary supplements" ( <code>group = dietary_supplements</code> ), "Vegetable oils and fat-and-oil products" ( <code>group = oil_fat</code> ), "Animal feed" ( <code>group = animal_feed</code> ), "Preserves" ( <code>group = preserves</code> ), "Veterinary medicines" ( <code>group = veterinary_medicines</code> ), "Sturgeon roe and salmon roe"

			(group = caviar), "Bicycles and bicycle frames" (group = bike), "Antiseptics and disinfectants" (group = antiseptics), "Food products (Groceries)" (group = grocery), "Cosmetics and household cleaning products" (group = cosmetics)
labels	Yes	Array of Strings	Array (list) of LCs applied to the item. <b>Limits:</b> 1. Only LCs of the Russian format (code type label_type = 7) with the status "Applied" (status = 47) or "Labeled items" (status = 50) are allowed. 2. Maximum 10,000 LCs per request.
params	Yes	Object	Manufacturing parameters. The set of parameters (additional characteristics) and conditions of their setting is determined by the belonging of the labeled item to the corresponding section of the catalog and can be retrieved by the /catalogs/{:code} method (see 3.2.2).
{:slug}	Yes/No	String, Integer, Decimal, Date, Boolean	The element key name corresponds to the key value (string) of the parameter specified in the slug attribute of the Param class object (see 4.1.5) listed in the mark_params array of the Category class object (see 4.1.3 and Table D.2 of Annex D). The element value depends on the type of parameter (additional characteristic)
<b>Request body example</b>			
<pre>{   "group": "dietary_supplements",   "labels": [     "0104812845008808215CjBmJL,pugAq\u001d91FFD0\u001d92dGVzdBSSkDJP1g/BAA bCv3wv/OaXx8Fb2u5JT9pZgKc=",     "0104812845008808215&amp;)'wXeauGKm\u001d91FFD0\u001d92dGVzdPBtdSpuGymZRA LebENUimTf271ALnTtHAC/ZFY=",     ...     "0104812845008808215j.FUYPHZM_nV\u001d91FFD0\u001d92dGVzdBs1wK4e8qW5oR XvEE8W49YCi+uoC4R3REf5730="   ],   "params": {     "manufacture_date": "2024-11-29",     "expiration_date": "2026-11-28"   } }</pre>			

**Table 3.5.14.4** HTTP response parameters, Code 200

Key	Data type	Description
report_id	Integer	Report ID
report_uid	String	UID, a unique identifier of the manufacturing

	parameter report (report_type = 14)
Response body example	
<pre>{   "report_id": 3684,   "report_uid": "fcacc009-a2ac-49b8-9271-71669eefec21" }</pre>	

### 3.5.15 Method "Owner change report"

#### 3.5.15.1. Representation A

Request type: POST  
Request string: /v2/reports/addChangeOwnerMade  
Parameters:

**Table 3.5.15.1** HTTP request parameters

Key	Required	Data type	Description
reason	Yes	String	Owner change reason. Enumeration value code change_of_owner obtained by /directories/{:code} method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.7</a> ). <b>Note:</b> The parameter value determines the set and the necessity of setting individual attributes.
group	Yes/No	String	Product group (the list of values is given in Table <a href="#">4.2.1.2</a> ). <b>Note:</b> Only required if the owner of the labeled item is changed (reason = change_owner_goods). Otherwise not used.
agent	Yes/No	Integer	ID of the predecessor agent (former owner) that previously owned unused LCs, UCMs, PCs and/or PPDCs. <b>Notes:</b> 1. The value of the id requisite of the Agent class object is specified (see <a href="#">4.1.2</a> ). 2. Must have the flag of a registered circulation participant (is_verified = True). <b>Note:</b> Only required if the owner change reason is business entity reorganization (reason = reorganization). Otherwise not used.
document_type	Yes/No	String	Type of underlying document. Enumeration value code shippingDocs obtained by /directories/{:code} method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.3</a> ). Allowed values: aktreorg: deed of conveyance; ttn: waybill (CMR); tnttn: delivery slip;

			<p>ettn: electronic waybill (eCMR); etn: electronic delivery slip. <b>Note:</b> Required if the owner change reason is business entity is <b>reason = reorganization.</b> May be filled in in other cases.</p>
document_date	Yes/No	Date	<p>Date of underlying document. <b>Format:</b> YYYY-MM-DD. <b>Note:</b> Required if the owner change reason is business entity is <b>reason = reorganization.</b> May be filled in in other cases.</p>
document_number	No	String	Number of underlying document
labels	Yes	Array of Strings	<p>Array of LCs or series and numbers of forms (labels) to be transferred to the new owner. <b>Limits:</b> 1. Maximum 100,000 LCs or labels per request; 2. LCs of labels must be of the same type LabelType (see <a href="#">4.2.2</a>); 3. In case of owner change due to business entity reorganization (<b>reason = reorganization</b>): 3.1. LCs or labels must belong to the previous owner listed in the agent attribute; 3.2. LCs or labels must not be used for labeling of goods or discarded (<b>status.code &lt; 50</b>); 4. In case of change of the owner of the labeled item (<b>reason = change_owner_goods</b>): 4.1. LCs or labels must belong to the same product group Group (see <a href="#">4.2.1</a>); 4.2. LCs or labels are used for labeling of goods and are in circulation (<b>status.code = 50 or 55</b>).</p>
comment	No	String	Comment, note
<b>Request body example</b>			
<b>Option 1.</b> Owner change in case of business entity reorganization ( <b>reason = reorganization</b> )			
<pre>{   "reason": "reorganization",   "agent": 10564,   "document_type": "actreorg",   "document_number": "5",   "document_date": "2024-09-30",   "labels": [     "0104813420015020212viLoMo1TcYrk\u001d910005\u001d92JFckw6KN9mYbzCVY3     TMQqvh6V586x5rreGd4fQvG33TshvyXnnKoZfoiGfwA5X8h2Zyw8HBXPLoBtS9wwAswZAw4",     "01048116440080642122urA5V4fFxZN\u001d910005\u001d92eTob8nQhDGvJKmXCU     L1TKmhJvFJdxomU1eDiXskLPRem",   ] }</pre>			

```

"01048101053767732121SN5tsFdXpXF\u001d910005\u001d92KaE6ufC45Py8NNYC4
CCLLb74zyJqHP5NQgdoeRhsYA6d",
...
"AAA012345289",
"AAA012345290"
]
}

```

**Option 2.** Change of the owner of the labeled item  
(reason = change\_owner\_goods)

```

{
  "reason": "change_owner_goods",
  "group": "shoes",
  "labels": [
    "0104813420015020212KWrkJTRRcLcH\u001d910005\u001d92xD6TNTg4mviY2AWZnD
F1NKNvK4ARuV8FsdLRAL6iBF5pVUm9MyJZ4gN9GBNB1rEHoxBFpcyyUnzp2vT4nojZbqd5",
    "0104813420015020212Q9GxPZWCw3ah\u001d910005\u001d92tUnujbqaRwfrEz7dyM
vtymQW3RTmsz6qZiDnmCnxuW7MCoalX8nF16b7C4JeRqGcnFu2ku3PBY4XNmr5ctsHfkTk",
    ...
    "0104813420015020212UTH6SMS6w4KA\u001d910005\u001d92mC17X7xGcDdSTCTNaL
YxWHwJ2oC1LJzgpsFFEq4N1dnGUeY2nbAT5F8UuSJVdEV4kf8b7ERT186sfPs85aZwMKsP "
  ],
  "comment": ""
}

```

**Table 3.5.15.2** HTTP response parameters, Code 200

Key	Data type	Description
report_id	String	UID, a unique identifier of the owner change report (report_type = 15)
<b>Response body example</b>		
<pre> {   "report_id": "7ce6e142-7786-48f9-b99c-cbb25446ad73" } </pre>		

### 3.5.15.2. Representation B

Request type: POST  
Request string: /v3/reports/addChangeOwnerMade  
Parameters:

**Table 3.5.15.3** HTTP request parameters

Key	Required	Data type	Description
reason	Yes	String	Owner change reason. Enumeration value code change_of_owner obtained by /directories/{:code} method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.7</a> ). <i>Note:</i> The parameter value determines the set and the necessity of setting individual attributes.
group	Yes/No	String	Product group (the list of values is given in Table <a href="#">4.2.1.2</a> ). <i>Note:</i> Only required if the owner of the labeled item is changed (reason =

			change_owner_goods). Otherwise not used.
agent	Yes/No	Integer	ID of the predecessor agent (former owner) that previously owned unused LCs, UCMs, PCs and/or PPDCs. <b>Notes:</b> 1. The value of the <code>id</code> requisite of the <code>Agent</code> class object is specified (see <a href="#">4.1.2</a> ). 2. Must have the flag of a registered circulation participant ( <code>is_verified = True</code> ). <b>Note:</b> Only required if the owner change reason is business entity reorganization ( <code>reason = reorganization</code> ). Otherwise not used.
document_type	Yes/No	String	Type of underlying document. Enumeration value code <code>shippingDocs</code> obtained by <code>/directories/{:code}</code> method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.3</a> ). Allowed values: <code>aktreorg</code> : deed of conveyance; <code>ttn</code> : waybill (CMR); <code>tnttn</code> : delivery slip (TN); <code>ettn</code> : electronic waybill (eTTN); <code>etn</code> : electronic delivery slip (eTN). <b>Note:</b> Required if the owner change reason is business entity is <code>reason = reorganization</code> . May be filled in in other cases.
document_date	Yes/No	Date	Date of underlying document. <b>Format:</b> YYYY-MM-DD. <b>Note:</b> Required if the owner change reason is business entity is <code>reason = reorganization</code> . May be filled in in other cases.
document_number	No	String	Number of underlying document
labels	Yes/No	Array of Strings	Array of LCs to be transferred to the new owner. <b>Limits:</b> 1. Maximum 100,000 LCs or labels per request; 2. LCs of labels must be of the same type <code>LabelType</code> (see <a href="#">4.2.2</a> ); 3. In case of owner change due to business entity reorganization ( <code>reason = reorganization</code> ): 3.1. LCs or labels must belong to the previous owner listed in the <code>agent</code> attribute; 3.2. LCs or labels must not be used for labeling of goods or discarded ( <code>status.code &lt; 50</code> );

			4. In case of change of the owner of the labeled item (reason = change_owner_goods): 4.1. LCs or labels must belong to the same product group Group (see 4.2.1); 4.2. LCs or labels are used for labeling of goods and are in circulation (status.code = 50 or 55). <b>Note:</b> Must be set if aggregates is not specified
aggregates	Yes/No	Array of Strings	Array of ACs to be transferred to the new owner. <b>Note:</b> Must be set if labels is not specified
comment	No	String	Comment, note
<b>Request body example</b>			
<b>Option 1.</b> Owner change in case of business entity reorganization (reason = reorganization)			
<pre>{   "reason": "reorganization",   "agent": 10564,   "document_type": "actreorg",   "document_number": "1001/1",   "document_date": "2025-01-10",   "labels": [     "0104812845008808215?&amp;T.nniB,0&amp;u\u001d91FFD0\u001d92dGVzdLRHvQ6LVjFS4H     woxLFJrFwQsXDTXbpz4bEcIRA=",     "0104812845008808215&amp;')WXeauGKm\u001d91FFD0\u001d92dGVzdPBtdSpuGymZRA     LebENUimTf271ALnTtHAC/ZFY=",     ...     "0104812845008808215(ne1bkaQptB?\u001d91FFD0\u001d92dGVzdEgtw1La2hx40B     lz73AQU2EbpVHvAvmeknf9+YM="   ],   "comment": "Переход ИП в созданное им юл" }</pre>			
<b>Option 2.</b> Change of the owner of the labeled item (reason = change_owner_goods)			
<pre>{   "reason": "change_owner_goods",   "group": "dietary_supplements",   "labels": [     "0100625273050882215&lt;AID7N0Td05T\u001d91FFD0\u001d92dGVzdBUtb1+pU+1oaj     FxrtPOKkvMoaRYA0cQyD+IWec=",     "0100625273050882215JKLDGHJT(34Q\u001d91FFD0\u001d92dGVzd06uV4PpGZGxDq     ZMgE++H9WeH7vMnBbvv1MGCHg=",     ...     "0100625273050882215XmFKi)t.pqB5\u001d91FFD0\u001d92dGVzdPSQ8VqE0fx8Ua     rPuutnIL6Y2LvykFo8w9gwP14="   ] }</pre>			

**Note:** The request must always contain at least one of the parameters in which the codes are listed ("labels" for LCs or "aggregates" for ACs).

**Table 3.5.15.4 HTTP response parameters, Code 200**

Key	Data type	Description
report_id	Integer	Report ID
report_uid	String	UID, a unique identifier of the owner change report (report_type = 15)
Response body example		
<pre>{   "report_id": 5321,   "report_uid": "c364a0ac-9458-41d7-9aed-8c63d9b957af" }</pre>		

**3.5.16 Method "Report on deletion of codes from the aggregate"**

Request type: POST  
Request string: /v3/reports/aggregateDelete  
Parameters:

**Table 3.5.16.1 HTTP request parameters**

Key	Required	Data type	Description
labels	Yes/No	Array of Strings	Array of LCs to be excluded from aggregate. <b>Limit:</b> Maximum 100,000 LCs per 1 query, including total for the entire aggregation tree (nesting structure including child LCs). <b>Notes:</b> 1. LCs must belong to the same product group and have the same status. 2. Must be set if aggregates is not specified
aggregates	Yes/No	Array of Strings	Array of ACs to be excluded from the upper level aggregate. <b>Note:</b> Must be set if labels is not specified
destroy_all_boxes	Yes	Boolean	Flag of destruction of the upper aggregate (i.e., the AC from which the codes are deleted): False: do not destroy the aggregation; True: destroy the aggregation.
Request body example (fragment)			
<pre>{   " labels": [     "0104812845008808215CjBmJL , pugAq\u001d91FFD0\u001d92dGVzdBSSkDJP1g/BAA     bCv3wv/OaXx8Fb2u5JT9pZgKc=",     "0104812845008808215&amp;')WXeauGKm\u001d91FFD0\u001d92dGVzdPBTdSpuGymZRA     LebENUimTf271ALnTtHAC/ZFY=",     ...     "0104812845008808215j.FUYPHZM_nV\u001d91FFD0\u001d92dGVzdBs1wK4e8qW5oR     XvEE8W49YCi+uoC4R3REf5730="   ],   "destroy_all_boxes": false }</pre>			

**Note:** The request must always contain at least one of the parameters in which the codes are listed ("labels" for LCs or "aggregates" for ACs).

**Table 3.5.16.2** HTTP response parameters, Code 200

Key	Data type	Description
report_id	Integer	Report ID
report_uid	String	UID, a unique identifier of the report on deletion of codes from aggregate (report_type = 16)
Response body example		
<pre>{   "report_id": 5459,   "report_uid": "eb1e7507-aa31-4021-ba8e-f6c4b4331d9a" }</pre>		

### 3.6 Agent handling methods

#### 3.6.1 Method "Agent details"

Request type: POST  
Request string: /contracts/agent  
Parameters:

**Table 3.6.1.1** HTTP request parameters

Key	Required	Data type	Description
unp	Yes	String	Taxpayer's Number or other equivalent identifier of the agent
Request body example			
<pre>{   "unp": "700016217" }</pre>			

**Table 3.6.1.2** HTTP response parameters, Code 200

Key	Data type	Description
	Object	Agent details. Agent class object (see <a href="#">4.1.2</a> )
Response body example		
<pre>{   "id": 176,   "name": "Открытое акционерное общество \"Белшина\"",   "unp": "700016217",   "country": "112",   "address": "Могилевская обл., г. Бобруйск, ш. Минское, д. 4",   "gln": "4811644900009",   "is_verified": true }</pre>		

#### 3.6.2 Method "Adding a new agent"

Request type: PUT  
Request string: /contracts/addAgent  
Parameters:

**Table 3.6.2.1 HTTP request parameters**

Key	Required	Data type	Description
country	Yes	String	Digital country code. Enumeration value code country obtained by /directories/{:code} method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.1</a> )
unp	Yes	String	Taxpayer's Number or other equivalent identifier of the agent
name	Yes	String	Agent name
address	No	String	Agent address
gln	No	String	Agent GLN
Request body example			
<pre>{   "name": "000 \"УНИВЕРСАЛ-ТРЕЙДИНГ\"",   "unp": "0274068089",   "address": "ул. Милашенкова, д. 4 корп. 7, г. Москва",   "country": "643",   "gln": "4630019409950" }</pre>			

**Table 3.6.2.2 HTTP response parameters, Code 200**

Key	Data type	Description
	Object	Agent details. Agent class object (see <a href="#">4.1.2</a> )
Response body example		
<pre>{   "id": 18413,   "name": "000 \"УНИВЕРСАЛ-ТРЕЙДИНГ\"",   "unp": "0274068089",   "country": "643",   "address": "ул. Милашенкова, д. 4 корп. 7, г. Москва",   "gln": "4630019409950",   "is_verified": false }</pre>		

### 3.6.3 Method "Agent list"

Request type: POST  
Request string: /contracts/agentsList  
Parameters:

**Table 3.6.3.1 HTTP request parameters**

Key	Required	Data type	Description
country	No	String	Digital country code. Enumeration value code country obtained by /directories/{:code} method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.1</a> )
unp	No	String	Taxpayer's Number or other equivalent identifier of the agent
name	No	String	Full name or part of the name of the agent. <b>Note:</b> Selection is made by substring search

show_disabled	No	Boolean	<p>Agent search mode:</p> <p>False: search only for valid agents, i.e., active or "enabled" (<code>status.code = 1</code>) records (default mode);</p> <p>True: search for all agents including blocked ones, i.e., including archived and disabled (<code>status.code = 0</code>) records.</p> <p><b>Note:</b> If the parameter is not specified, the default search is performed in <code>show_disabled = False</code> mode.</p>
<b>Request body example</b>			
<pre>{   "unp": "500040357",   "country": "112" }</pre>			

**Table 3.6.3.2** HTTP response parameters, Code 200

Key	Data type	Description
agents_list	Array of Objects	<p>Array containing information about agents.</p> <p>Agent class objects (see <a href="#">4.1.2</a>).</p> <p><b>Note:</b> The first 500 records that meet the selection criteria are returned.</p>
<b>Response body example (fragment)</b>		
<pre>{   "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",       "status": {         "code": 1,         "message": "Включен"       },       "is_verified": true     },     {       "id": 18708,       "name": "Щучинский филиал Открытое акционерное общество \"Молочный Мир\"",       "unp": "500040357",       "country": {         "code": "112",         "name": "Беларусь",         "comment": null       },       "address": "Гродненская область, г. Щучин, ул. 17 сентября, 45",       "gln": "4810099900077",       "status": {</pre>		

```

        "code": 1,
        "message": "Включен"
    },
    "is_verified": true
},
...
{
    "id": 22409,
    "name": "Производственный цех Щучинского филиала ОАО \"Молочный
Мир\"",
    "unp": "500040357",
    "country": {
        "code": "112",
        "name": "Беларусь",
        "comment": null
    },
    "address": "г. Скидель",
    "gln": "4810099900039",
    "status": {
        "code": 1,
        "message": "Включен"
    },
    "is_verified": false
}
]
}

```

### 3.7 Shipment handling methods

#### 3.7.1 Method "Code shipment"

##### 3.7.1.1. Representation A

Request type: PUT  
Request string: /ships/add  
Parameters:

**Table 3.7.1.1 HTTP request parameters**

Key	Required	Data type	Description
shipping_doc	Yes	String	Type of accompanying document. Enumeration value code shippingDocs obtained by /directories/{:code} method (see 3.8.3; the list of values is given in Table 4.3.3)
nomer_tn	Yes	String	Number of accompanying document; may also include series
agent	Yes	Integer	ID of the recipient counterparty to which the codes are shipped and which will have to perform their capitalization. <b>Note:</b> The value of the id

			requisite of the Agent class object is specified (see <a href="#">4.1.2</a> )
count	Yes	Integer	Total number of units of labeled goods according to the underlying document
comment	No	String	Shipment comment
country	Yes	String	Digital code of the country to which the shipment is being made. Enumeration value code country obtained by /directories/{:code} method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.1</a> )
price	No	Decimal	Total cost of labeled goods according to the underlying document
currency	No	String	Shipment currency Enumeration value code currency obtained by /directories/{:code} method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.2</a> )
operation_date	Yes	Date	Date of shipment (date of the participant's operation, usually corresponds to the date of the accompanying document). <b>Format:</b> YYYY-MM-DD. <b>Limit:</b> The shipment date must not be earlier than the LC generation date
labels	Yes	Array of Strings	Array of LCs or ACs to be shipped <b>Note:</b> ACs can only be used for shipments within the Republic of Belarus. <b>Limits:</b> The codes listed in the array must meet the following conditions: 1. LCs must belong to the same product group; 2. LC must not be included into another aggregate; 3. The AC must not be empty, i.e., it must contain nested codes (child or lower nesting level LCs or ACs); 4. AC must not be included into the top level aggregate; 5. LCs and ACs cannot be specified at the same time; 6. The total number of LCs (including those included in aggregation) per shipment:

			6.1. For shipments to EAEU countries: maximum 30,000 codes; 6.2. In other cases: maximum 100,000 codes. 7. For shipments from the Republic of Belarus, ACs are not sent; only LCs must be specified (taking into account item 6).
eas_products	No	Object	Additional information about the goods to be supplied. <b>Note:</b> Used in case of transfer of shipment information in cross-border trade between the EAEU member states.
{:gtin}	Yes	String	The element key name corresponds to the GTIN value of the item. The element value contains additional parameters of shipment by the specified GTIN
product_cost	No	String, Integer	Unit price (including VAT, if the goods are subject to VAT). <i>See the <a href="#">note</a> below.</i>
product_tax	No	String, Integer	Total unit VAT. <i>See the <a href="#">note</a> below.</i>
certificate_document_data	No	Array of Objects	Array of information on documents confirming compliance of goods with quality and/or safety requirements
certificate_type	Yes	String	Type of document confirming compliance (authorization document). Enumeration value code <code>certificate_types</code> retrieved from <code>/directories/{:code}</code> method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.6</a> , acceptable values are given in Annex <a href="#">F</a> )
certificate_number	Yes	String	Number of the document confirming compliance
certificate_date	Yes	Date	Date of the document confirming compliance. <b>Format:</b> YYYY-MM-DD
<b>Request body example (fragment)</b>			
<pre>{   "shipping_doc": "tnttn",   "nomer_tn": "0A03456789",   "agent": 18413,   "count": 10,</pre>			

```

"comment": "Отгрузка согласно счёт-фактуре № 12456 от 04.11.2020",
"country": "643",
"price": 4567.89,
"currency": "rub643",
"operation_date": "2020-11-24",
"labels": [
  "0104811309016724212R7V7577\u001d93iv4=",
  "01048113090167242128v8LZVn\u001d936rX0",
  ...
  "0104811309016724212f4DSzvd\u001d936Dyt"
],
"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	Information message
ship	Integer	Shipment number (identifier)
Response body example		
<pre> {   "message": "#4355 успешно создана.",   "ship": 4355 } </pre>		

### 3.7.1.2. Representation B

Request type: POST  
Request string: /v3/ships/add  
Parameters:

**Table 3.7.1.3** HTTP request parameters

Key	Required	Data	Description
-----	----------	------	-------------

		type	
group	Yes	String	Product group (the list of values is given in Table <a href="#">4.2.1.2</a> )
shipping_doc	Yes	String	Type of accompanying document. Enumeration value code <code>shippingDocs</code> obtained by <code>/directories/{:code}</code> method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.3</a> )
nomer_tn	Yes	String	Number of accompanying document; may also include series
country	Yes	String	Digital code of the country to which the shipment is being made. Enumeration value code <code>country</code> obtained by <code>/directories/{:code}</code> method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.1</a> )
agent	Yes	Integer	ID of the recipient counterparty to which the codes are shipped and which will have to perform their capitalization. <b>Note:</b> The value of the <code>id</code> requisite of the <code>Agent</code> class object is specified (see <a href="#">4.1.2</a> )
count	Yes	String	Total number of units of labeled goods according to the underlying document
comment	No	String	Shipment comment
price	No	String	Total cost of labeled goods according to the underlying document
currency	No	String	Shipment currency Enumeration value code <code>currency</code> obtained by <code>/directories/{:code}</code> method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.2</a> )
operation_date	Yes	Date	Date of shipment (date of the participant's operation, usually corresponds to the date of the accompanying document). <b>Format:</b> YYYY-MM-DD. <b>Limit:</b> The shipment date must not be earlier than the LC generation date
eas_products	Yes/No	Object	Additional information about the goods to be supplied. <b>Notes:</b> 1. Used in case of transfer of

			shipment information in cross-border trade between the EAEU member states. 2. Required for the product group "Veterinary medicines" (group = <code>veterinary_medicines</code> ).
<code>{:gtin}</code>	Yes	String	The element key name corresponds to the GTIN value of the item. The element value contains additional parameters of shipment by the specified GTIN. <i>Note:</i> The element must in any case contain nested attributes "product_cost" and "product_currency" or "certificate_document_data"
<code>product_cost</code>	Yes/No	String	Unit price (including VAT, if the goods are subject to VAT). <i>See note below (format 1).</i>
<code>product_tax</code>	No	String	Total unit VAT. <i>See note below (format 1).</i>
<code>product_currency</code>	Yes/No	String	Product price currency. Enumeration value code <code>currency</code> obtained by <code>/directories/{:code}</code> method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.2</a> ). <i>Note:</i> Required if the "product_cost" attribute contains a unit price.
<code>certificate_document_data</code>	Yes/No	Array of Objects	Array of information on documents confirming compliance of goods with quality and/or safety requirements or registration of goods in accordance with the established procedure. <i>Note:</i> Required for the product group "Veterinary medicines" (group = <code>veterinary_medicines</code> ).
<code>certificate_type</code>	Yes	String	Type of document confirming compliance (authorization document). Enumeration value code <code>certificate_types</code> retrieved from method <code>/directories/{:code}</code> (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.6</a> , acceptable values are given in

			Annex F). <b>Note:</b> For the product group "Veterinary medicines" (group = <code>veterinary_medicines</code> ), information on the registration certificate (certificate_type = <code>REGISTRATION_VET_CERTIFICATE</code> ) must be specified
certificate_number	Yes	String	Number of the document confirming compliance
certificate_date	Yes	Date	Date of the document confirming compliance. <b>Format:</b> YYYY-MM-DD
labels	Yes/No	Array of Strings	Array of LCs to be shipped. <b>Limits:</b> The codes listed in the array must meet the following conditions: 1. LCs must belong to the same product group; 2. Total number of LCs per 1 shipment: 2.1. For shipments to EAEU countries: maximum 30,000 codes; 2.2. In other cases: maximum 100,000 codes. <b>Note:</b> Must be set if <code>aggregates</code> is not specified
aggregates	Yes/No	Array of Strings	AC array to be shipped. <b>Limits:</b> The codes listed in the array must meet the following conditions: 1. The AC must not be empty, i.e., it must contain nested codes (child or lower nesting level LCs or ACs); 2. ACs (including nested ones) must contain LCs belonging to the same product group; 3. Total number of LCs included in the aggregation (taking into account child nesting levels) per 1 shipment: 3.1. For shipments to EAEU countries: maximum 30,000 codes; 3.2. In other cases: maximum 100,000 codes. 4. For shipments from the Republic of Belarus, ACs are

			<p>not sent; only LCs listed in the <code>labels</code> array must be specified.</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. Must be set if <code>labels</code> is not specified.</li> <li>2. ACs can only be used for shipments within the Republic of Belarus.</li> </ol>
--	--	--	--

**Request body example (fragment)**

```
{
  "group": "dietary_supplements",
  "shipping_doc": "tnttn",
  "nomer_tn": "БГ0100025",
  "country": "643",
  "agent": 18413,
  "contract": null,
  "count": "5",
  "comment": "",
  "price": "10253.75",
  "currency": "rub643",
  "operation_date": "2024-11-30",
  "eas_products": {
    "04812845008808": {
      "product_cost": "2050.75",
      "product_tax": "205.08",
      "product_currency": "rub643",
      "certificate_document_data": [
        {
          "certificate_type": "STATE_REGISTRATION_CERTIFICATE",
          "certificate_number": "3",
          "certificate_date": "2024-11-29"
        }
      ]
    }
  ],
  "labels": [
    "0104812845008808215CjBmJL,pugAq\u001d91FFD0\u001d92dGVzdBSSkDJP1g/BAA
bCv3wv/OaXx8Fb2u5JT9pZgKc=",
    "0104812845008808215&)' )WXeauGKm\u001d91FFD0\u001d92dGVzdPBtdSpuGymZRA
LebENUimTf271ALnTtHAC/ZFY=",
    ...
    "0104812845008808215j.FUYPHZM_nV\u001d91FFD0\u001d92dGVzdBs1wK4e8qW5oR
XvEE8W49YCi+uoC4R3REf5730="
  ]
}
```

**Note:** The request must always contain at least one of the parameters in which the codes are listed ("`labels`" for LCs or "`aggregates`" for ACs).

**Table 3.7.1.4 HTTP response parameters, Code 200**

Key	Data type	Description
<code>ship_id</code>	Integer	Shipment number (identifier)
<b>Response body example</b>		

```
{
  "ship_id": 2000040067
}
```

**Note:** Additional information on unit price (attribute "product\_cost") and total unit VAT (attribute "product\_tax") apply in case of shipment to the Russian Federation and is sent:

1. On voluntary basis, including within contractual relations between the exporter (seller, shipper) and importer (buyer, recipient).
2. In Russian rubles (RUB).
3. In one of the following formats:

*Format 1.* In rubles as a number with two decimal places (digits) represented 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 Method "Shipment details"

Request type: GET  
Request string: /ships/{:code}  
Parameters:

**Table 3.7.2.1 HTTP request parameters**

Key	Required	Data type	Description
code	Yes	Integer	Shipment number. Shipment ID retrieved from /ships/add (see 3.7.1.1), /ships/list or /ships/shipped method (see 3.7.5.1)
<b>Request string example</b>			
/ships/4355			

**Table 3.7.2.2 HTTP response parameters, Code 200**

Key	Data type	Description
	Object	Shipment details. Ship class object (see 4.1.7)
<b>Response body example</b>		
<pre>{   "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,   "taken_at": "2020-11-24 16:47:02+03",   "received_count": null,   "price": {     "price": 4567.89,     "currency": "rub643"   },   "contract": false, }</pre>		

```
"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": "С000 \"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 Method "Shipped code list"

#### 3.7.3.1. Representation A

Request type: POST  
Request string: /ships/labels  
Parameters:

**Table 3.7.3.1 HTTP request parameters**

Key	Required	Data type	Description
code	Yes	Integer	Shipment number. Shipment ID retrieved from /ships/add (see <a href="#">3.7.1.1</a> ), /ships/list or /ships/shipped method (see <a href="#">3.7.5.1</a> )
<b>Request body example</b>			
<pre> {   "code": 4355 } </pre>			

**Table 3.7.3.2 HTTP response parameters, Code 200**

Key	Data type	Description
aggregates	Array of Objects	Array of all ACs (indicating parent ACs, if any)
label	String	AC
status	Object	Current AC status. LabelStatus class object (see <a href="#">4.2.3</a> )
parent	String	Upper level AC, which includes the current AC
labels	Array of Objects	Array of all LCs (indicating parent ACs, if any)
label	String	LC
status	Object	Current LC status LabelStatus class object (see <a href="#">4.2.3</a> )
snumber	String	Series and number of a document with a certain level of protection. <i>Note:</i> Used for code types with the flag <code>is_bso = True</code>
parent	String	Upper level AC, which includes the current LC
<b>Response body example (fragment)</b>		
<pre> {   "aggregates": [ </pre>		

```

    {
      "label": "00048116129000000529",
      "status": {
        "code": 30,
        "message": "Валидирован"
      },
      "parent": ""
    },
    ...
    {
      "label": "000481901888000002502048113090167243761320111230987zyx",
      "status": {
        "code": 30,
        "message": "Валидирован"
      },
      "parent": "00048116129000000529"
    }
  ],
  "labels": [
    {
      "label": "0104810268018930212Z6hrpa\u001d93Y!&m",
      "status": {
        "code": 50,
        "message": "Промаркирован"
      },
      "snomer": null,
      "parent": "00048190188800000250204811309016724376\u001d1320111230
123abc"
    },
    ...
    {
      "label": "0104810268018930212CnSujHX\u001d93BiYG",
      "status": {
        "code": 50,
        "message": "Промаркирован"
      },
      "snomer": null,
      "parent": "00048190188800000180204811309016724376\u001d1320111230
987zyx"
    }
  ]
}

```

### 3.7.3.2. Representation B

Request type: POST  
Request string: /v3/ships/downloads  
Parameters:

**Table 3.7.3.3** HTTP request parameters

Key	Required	Data type	Description
ship_id	Yes	Integer	Shipment number. Shipment ID retrieved from /v3/ships/add (see <a href="#">3.7.1.1</a> ), /v3/ships/list or /v3/ships/shipped method (see <a href="#">3.7.5.2</a> )

file_type	Yes	String	Type of information returned: labels_sent: sent LCs; labels_taken: received LCs; labels_error: incorrect codes (LCs and/or ACs); aggregates_sent: sent ACs; aggregates_taken: received ACs; destroy_aggregates: ACs destroyed after shipment; destroy_aggregates_after_ship: ACs destroyed after acceptance.
<b>Request body example</b>			
<pre>{   "ship_id": 2000040045,   "file_type": "labels_sent" }</pre>			

**Table 3.7.3.4** HTTP response parameters, Code 200

Key	Data type	Description
ship	Integer	Shipment number (identifier)
file_type	String	Type of information returned
codes	Array of Strings	Array (list) of LCs or ACs
detail	String	Information message
<b>Response body example (fragment)</b>		
<i>Option 1. Code information found</i>		
<pre>{   "ship_id": 2000040045,   "file_type": "labels_sent",   "codes": [     "0105201409805158215cFp&lt;eKzEGR;f\u001d91FFD0\u001d92dGVzdI3PoF8ZiWLNW u55wIx04qjEU+aAdiMeEo7s2hE=",     ...   ] }</pre>		
<i>Option 2. Requested code information is not available</i>		
<pre>{   "ship_id": 2000040045,   "file_type": "labels_taken",   "detail": "файл не найден" }</pre>		

### 3.7.4 Method "Code capitalization"

#### 3.7.4.1. Representation A

Request type: POST  
Request string: /ships/take  
Parameters:

**Table 3.7.4.1** HTTP request parameters

Key	Required	Data type	Description
code	Yes	Integer	Shipment number. Shipment ID retrieved

			from method <code>/ships/list</code> (see <a href="#">3.7.5.1</a> )
labels	Yes	Array of Strings	<p>Array of LCs of ACs to be accepted. Can be retrieved using method <code>/ships/labels</code> (see <a href="#">3.7.3.1</a>).</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. Performed only once.</li> <li>2. Array <code>labels</code> contain the codes actually received.</li> <li>3. Partial acceptance is possible; in this case all non-accepted codes from the specified shipment that were not listed in array <code>labels</code> are automatically returned to the shipper.</li> <li>4. If array <code>labels</code> is empty, the entire shipment will be rejected and the codes included in it will be automatically returned to the shipper.</li> </ol>
<b>Request body example (fragment)</b>			
<pre>{   "code": 4355,   "labels": [     "0104811309016724212R7V7577\u001d93iv4=",     "01048113090167242128v8LZVn\u001d936rX0",     ...     "0104811309016724212f4DSzvd\u001d936Dyt"   ] }</pre>			

**Table 3.7.4.2** HTTP response parameters, Code 200

Key	Data type	Description
message	String	Information message
count	Integer	Total number of capitalized codes (including LCs and ACs at all nesting levels)
<b>Response body example</b>		
<pre>{   "message": "Успешно оприходовано",   "count": 15 }</pre>		

### 3.7.4.2. Representation B

Request type: POST  
Request string: `/v3/ships/take`  
Parameters:

**Table 3.7.4.3** HTTP request parameters

Key	Required	Data type	Description
ship_id	Yes	Integer	Shipment number. Shipment ID retrieved from method <code>/v3/ships/list</code> (see <a href="#">3.7.5.2</a> )
labels	Yes/No	Array of Strings	Array of LCs to be accepted. Can be

			retrieved using method /v3/ships/downloads (see <a href="#">3.7.3.2</a> ). <b>Note:</b> Specified in case of acceptance by LC
aggregates	Yes/No	Array of Strings	AC array to be accepted. Can be retrieved using method /v3/ships/downloads (see <a href="#">3.7.3.2</a> ). <b>Note:</b> Specified in case of acceptance by AC
Request body example (fragment)			
<pre>{   "ship_id": 2000040045,   "labels": [     "0105201409805158215cFp&lt;eKzEGR;f\u001d91FFD0\u001d92dGVzdI3PoF8ZiWLNW u55wIx04qjEU+aAdiMeEo7s2hE=",     ...   ] }</pre>			

**Notes:**

1. A request must always contain at least one of the parameters listing the accepted codes ("labels" for LCs or "aggregates" for ACs) or an empty array if all the corresponding codes are rejected.
2. Performed only once.
3. The "labels" array or the "aggregates" array lists the actually accepted codes (LCs or ACs, respectively).
4. Partial acceptance is possible; in this case all non-accepted codes from the specified shipment that were not listed in the "labels" array or "aggregates" array are automatically returned to the shipper.
5. In case the "labels" array or the "aggregates" array is not completed, the entire shipment related to the LCs or ACs, respectively, will be rejected and the transmitted codes will be automatically returned to the shipper.

**Table 3.7.4.4** HTTP response parameters, Code 200

Key	Data type	Description
detail	String	Information message
ship_id	Integer	Number (identifier) of the shipment being accepted
Response body example		
<pre>{   "detail": "Приемка начата",   "ship_id": 2000040045 }</pre>		

**3.7.5 Methods "Shipment list"**

**3.7.5.1. Representation A**

Request type: POST  
Request string: shipper context (shipment)  
/ships/shipped

recipient context (acceptance)  
/ships/list

Parameters:

**Table 3.7.5.1 HTTP request parameters**

Key	Required	Data type	Description
date_from	Yes	Date	Interval start date. <b>Format:</b> YYYY-MM-DD
date_to	Yes	Date	Interval end date. <b>Format:</b> YYYY-MM-DD
status	No	Integer	Code (identifier) of shipment status (the list of values is given in Table 4.2.6.2)
country	No	String	Digital country code. Enumeration value code country obtained by /directories/{:code} method (see 3.8.3; the list of values is given in Table 4.3.1)
page	Yes	Integer	Shipment list page number
Request body example			
<pre>{   "date_from": "2020-11-20",   "date_to": "2020-11-24",   "status": 10,   "country": "643",   "page": 1 }</pre>			

**Table 3.7.5.2 HTTP response parameters, Code 200**

Key	Data type	Description
count	Integer	Total number of shipments in the sample with current selections (filters)
ships_list	Array of Objects	Array containing information about shipments. Ship class objects (see 4.1.7)
Response body example		
<pre>{   "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,       "taken_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": {</pre>		

```
        "name": "ТН - товарная накладная",
        "code": "tnttn"
    },
    "code": 4355,
    "status": {
        "code": 12,
        "message": "Отгружен"
    },
    "user_from": {
        "name": "С000 \"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": "000 \"УНИВЕРСАЛ-ТРЕЙДИНГ\"",
        "address": "ул. Милашенкова, д. 4 корп. 7, г. Москва",
        "unp": "0274068089",
        "gln": "4630019409950",
        "country": "643"
    },
    "country": {
        "id": "643",
        "name": "Россия"
    }
}
]
```

### 3.7.5.2. Representation B

Request type: GET  
Request string: shipper context (shipment)  
/v3/ships/shipped?[page={:page}]  
&[page\_size={:page\_size}]  
&[ordering={:ordering}]  
&[start\_date={:start\_date}]  
&[end\_date={:end\_date}]  
&[nomer\_tn={:nomer\_tn}]  
&[country={:country}]  
&[agent={:agent}]  
&[start\_operation\_date={:start\_operation\_date}]  
&[end\_operation\_date={:end\_operation\_date}]  
&[start\_received\_date={:start\_received\_date}]  
&[end\_received\_date={:end\_received\_date}]  
&[currency={:currency}]  
&[status={:status}]  
&[search={:search}]  
&[cancel\_by\_admin={:cancel\_by\_admin}]

```

recipient context (acceptance)
/v3/ships/list?[page={:page}]
&[page_size={:page_size}]
&[ordering={:ordering}]
&[start_date={:start_date}]
&[end_date={:end_date}]
&[nomer_tn={:nomer_tn}]
&[start_operation_date={:start_operation_date}]
&[end_operation_date={:end_operation_date}]
&[start_received_date={:start_received_date}]
&[end_received_date={:end_received_date}]
&[currency={:currency}]
&[status={:status}]
&[search={:search}]
&[cancel_by_admin={:cancel_by_admin}]

```

Parameters:

**Table 3.7.5.3** HTTP request parameters

Key	Required	Data type	Description
page	No	Integer	Shipment list page number
page_size	No	Integer	Number of entries per page. <i>Note:</i> Maximum 200, by default, up to 15 per page.
ordering	No	String	Data sorting conditions: created_at: by creation date and time in ascending order; - created_at: by creation date and time in descending order; id: by ascending shipment number (identifier); - id: by descending shipment number (identifier)
start_date	No	Date	Shipment creation start date. <i>Format:</i> YYYY-MM-DD
end_date	No	Date	Shipment creation end date. <i>Format:</i> YYYY-MM-DD
nomer_tn	No	String	Number of accompanying document; may also include series
country	No	String	Digital country code. Enumeration value code country obtained by /directories/{:code} method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.1</a> )
agent	No	Integer	ID of the recipient counterparty to which the codes are shipped. <i>Note:</i> The value of the id requisite of the Agent class object is specified (see <a href="#">4.1.2</a> )
start_operation_date	No	Date	Start date of shipment according to

			accompanying document. <b>Format:</b> YYYY-MM-DD
end_operation_date	No	Date	End date of shipment according to accompanying document. <b>Format:</b> YYYY-MM-DD
start_received_date	No	Date	Start date of acceptance. <b>Format:</b> YYYY-MM-DD
end_received_date	No	Date	End date of acceptance. <b>Format:</b> YYYY-MM-DD
currency	No	String	Shipment currency Enumeration value code currency obtained by /directories/{:code} method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.2</a> )
status	No	Integer	Code (identifier) of shipment status (the list of values is given in Table <a href="#">4.2.6.2</a> )
search	No	Integer	Shipment number. Shipment ID by which the list is searched
cancel_by_admin	No	Boolean	Flag of cancelled shipment: False: shipment cancelled by recipient; True: shipment cancelled by Operator.
<b>Request string example</b>			
<b>Option 1.</b> Selection by parameters			
/v3/ships/shipped?page=1&page_size=15&ordering=-created_at&start_date=2024-11-29&end_date=2024-11-30&country=643			
<b>Option 2.</b> Search by shipment number			
/v3/ships/list?search=2000039814			

**Table 3.7.5.4** HTTP response parameters, Code 200

Key	Data type	Description
next	String	Link to the next page of the shipment list. <b>Note:</b> Null if none.
previous	String	Link to the previous page of the shipment list. <b>Note:</b> Null if none.
results	Array of Objects	Array containing information about shipments. Ship class objects (see <a href="#">4.1.7</a> )
<b>Response body example (fragment)</b>		
<b>Option 1.</b> Selection by parameters		
<pre>{   "next": "http://api.datamark.by/v3/ships/shipped?country=643&amp;end_date=2024-11-30&amp;ordering=-created_at&amp;page=2&amp;page_size=15&amp;start_date=2024-11-29",   "previous": null,   "results": [     {       "id": 2000040067,       "name": "БГ0100025",       "count": 5,       "comment": "",       "created_at": "2024-11-30T21:36:48.936364",       "updated_at": "2024-11-30T21:36:49.226942",       "contract_id": null,       "agent_id": 18413,     }   ] }</pre>		

```
"reason": "regular",
"ship_id": "50bd16d8-22cf-4e2e-bde4-65b4281d677a",
"user_info_from": {
  "name": "C000 \"Свиток\"",
  "address": "г. Минск, ул. Тимирязева, д.2, ком. 16",
  "unp": "190704463",
  "gln": "4811612900000",
  "country": "112",
  "is_rules_agree": true
},
"user_info_to": null,
"agent_to": {
  "name": "000 \"УНИВЕРСАЛ-ТРЕЙДИНГ\"",
  "address": "ул. Милашенкова, д. 4 корп. 7, г. Москва",
  "unp": "0274068089",
  "gln": "4630019409950",
  "country": "643"
},
"shipping_doc": {
  "code": "tnttn",
  "name": "ТН – товарная накладная"
},
"country": {
  "code": "643",
  "name": "Россия"
},
"status": {
  "code": 17,
  "message": "Ожидает приемку"
},
"operation_date": "2024-11-30",
"taken_at": null,
"real_count": 5,
"aggregate_count": 0,
"received_count": null,
"items": [
  {
    "gtin": "04812845008808",
    "name": "Биологически активные добавки (БАД)",
    "count": 5,
    "group": "dietary_supplements",
    "tnved": "3002490001",
    "articul": null,
    "document_id": null
  }
],
"labels": [
  {
    "id": 7,
    "name": "Код стран ЕАЭС идентификации товара",
    "count": 12
  },
  {
    "id": 6,
    "name": "Код агрегации (SSCC)",
    "count": 0
  }
]
```

```
    ],
    "all_aggregates_count": 0,
    "files": {
      "labels_sent": "ship_id-2000040067-labels_sent-kYZhkh.csv",
      "destroy_aggregates": null
    },
    "price": 10253.75,
    "currency": "rub643",
    "nds_sum": 0.0,
    "gtins_prices": null
  },
  {
    "id": 2000040066,
    "name": "БГ0100024",
    "count": 5,
    "comment": "",
    "created_at": "2024-11-30T18:13:07.529190",
    "updated_at": "2024-11-30T20:12:14.259231",
    "contract_id": null,
    "agent_id": 18413,
    "reason": "regular",
    "ship_id": "f601006e-ffb2-4433-9861-ae60a7476dc5",
    "user_info_from": {
      "name": "СООО \"Свиток\"",
      "address": "г. Минск, ул. Тимирязева, д.2, ком. 16",
      "unp": "190704463",
      "gln": "4811612900000",
      "country": "112",
      "is_rules_agree": true
    },
    "user_info_to": null,
    "agent_to": {
      "name": "ООО \"УНИВЕРСАЛ-ТРЕЙДИНГ\"",
      "address": "ул. Милашенкова, д. 4 корп. 7, г. Москва",
      "unp": "0274068089",
      "gln": "4630019409950",
      "country": "643"
    },
    "shipping_doc": {
      "code": "tnttn",
      "name": "ТН – товарная накладная"
    },
    "country": {
      "code": "643",
      "name": "Россия"
    },
    "status": {
      "code": 20,
      "message": "Принят"
    },
    "operation_date": "2024-11-30",
    "taked_at": "2024-11-30T20:12:14.259231",
    "real_count": 5,
    "aggregate_count": 0,
    "received_count": null,
    "items": [
      {
```

```
        "gtin": "00625273050882",
        "name": "БИОЛОГИЧЕСКИ АКТИВНАЯ ДОБАВКА",
        "count": 5,
        "group": "dietary_supplements",
        "tnved": "3002490001",
        "articul": null,
        "document_id": null
    }
],
"labels": [
    {
        "id": 7,
        "name": "Код стран ЕАЭС идентификации товара",
        "count": 1
    },
    {
        "id": 6,
        "name": "Код агрегации (SSCC)",
        "count": 0
    }
],
"all_aggregates_count": 0,
"files": {
    "labels_sent": "ship_id-2000040066-labels_sent-fzLgqo.csv",
    "labels_taken": "ship_id-2000040066-labels_taken-KLaTHL.csv",
    "destroy_aggregates": null
},
"price": 0.0,
"currency": "rub643",
"nds_sum": 0.0,
"gtins_prices": null
},
...
]
```

**Option 2. Search by shipment number**

```
{
  "next": null,
  "previous": null,
  "results": [
    {
      "id": 2000040045,
      "name": "АБ0123456",
      "count": 1,
      "comment": "",
      "created_at": "2024-11-29T15:54:56.934466",
      "updated_at": "2024-11-29T15:54:57.193925",
      "contract_id": null,
      "agent_id": 21513,
      "reason": "regular",
      "ship_id": null,
      "user_info_from": {
        "name": "000 \"Витебск-Торг\"",
        "address": "г.Витебск ул.Витебская, 1",
        "unp": "300238339",
        "gln": "4812767900003",
        "country": "112",
      }
    }
  ]
}
```

```
        "is_rules_agree": true
    },
    "user_info_to": {
        "name": "C000 \"Свиток\"",
        "address": "г. Минск, ул. Тимирязева, д.2, ком. 16",
        "unp": "190704463",
        "gln": "4811612900000",
        "country": "112",
        "is_rules_agree": true
    },
    "agent_to": {
        "name": "C000 \"Свиток\"",
        "address": "г. Минск, ул. Тимирязева, д.2, ком. 16",
        "unp": "190704463",
        "gln": "4811612900000",
        "country": "112"
    },
    "shipping_doc": {
        "code": "tnttn",
        "name": "ТН – товарная накладная"
    },
    "country": {
        "code": "112",
        "name": "Беларусь"
    },
    "status": {
        "code": 10,
        "message": "В пути"
    },
    "operation_date": "2024-11-29",
    "taked_at": null,
    "real_count": 40,
    "aggregate_count": 0,
    "received_count": null,
    "items": [
        {
            "gtin": "05201409805158",
            "name": "Масло оливковое",
            "count": 40,
            "group": "oil_fat",
            "tnved": null,
            "articul": null,
            "document_id": null
        }
    ],
    "labels": [
        {
            "id": 7,
            "name": "Код стран ЕАЭС идентификации товара",
            "count": 1
        },
        {
            "id": 6,
            "name": "Код агрегации (SSCC)",
            "count": 0
        }
    ]
},
```

```

    "all_aggregates_count": 0,
    "files": {
      "labels_sent": "ship_id-2000040045-labels_sent-oxEopV.csv",
      "destroy_aggregates": null
    },
    "price": 0.0,
    "currency": "byn933",
    "nds_sum": 0.0,
    "gtins_prices": null
  }
]
}

```

### 3.7.6 Method "Diary product supply registration"

Request type: POST  
Request string: /ships/gtin  
Parameters:

**Table 3.7.6.1** HTTP request parameters

Key	Required	Data type	Description
group	Yes	String	Product group (the list of values is given in Table 4.2.1.2). <i>Note:</i> Available only for product group "Dairy products" (group = milk)
shipping_doc	Yes/No	String	Type of accompanying document. Enumeration value code shippingDocs obtained by /directories/{:code} method (see 3.8.3; the list of values is given in Table 4.3.3). <i>Note:</i> Required if the operation type is operation = export
operation	Yes	String	Operation type: <b>export:</b> export registration; <b>import:</b> import registration
document_id	Yes/No	String	Unique identifier of the document by which the acceptance is performed retrieved from /ships/gtinList method (see 3.7.7). <i>Note:</i> Required if the operation type is operation = import and the supply information is sent by the exporter and received in SIS "Electronic Mark".
nomer_tn	Yes/No	String	Number of accompanying document; may also include

			series. <b>Note:</b> Required if the operation type is <code>operation = export</code>
country	Yes/No	String	Digital code of the country to which the supply is being made. Enumeration value code <code>country</code> obtained by <code>/directories/{:code}</code> method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.1</a> ). <b>Note:</b> Required if the operation type is <code>operation = export</code>
agent	Yes/No	Integer	ID of the recipient agent to which the supply is made. <b>Notes:</b> 1. The value of the <code>id</code> requisite of the <code>Agent</code> class object is specified (see <a href="#">4.1.2</a> ). 2. Required if the operation type is <code>operation = export</code> .
comment	No	String	Supply comment
price	No	String	Total cost of goods supplied
currency	Yes/No	String	Supply currency. Enumeration value code <code>currency</code> obtained by <code>/directories/{:code}</code> method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.2</a> ). <b>Note:</b> Required if the operation type is <code>operation = export</code>
operation_date	Yes/No	Date	Date of supply (date of the participant's operation, usually corresponds to the date of the accompanying document). <b>Format:</b> YYYY-MM-DD. <b>Note:</b> Required if the operation type is <code>operation = export</code>
gtin	Yes	Object	Parameters of volume and grade accounting of supplied goods in terms of GTIN and quantity
<code>{:gtin}</code>	Yes	String	The element key name corresponds to the GTIN value of the item. The element value contains supply parameters by the specified GTIN. <b>Template:</b> <code>[0-9]{14}</code>
count	Yes	Integer	Quantity of goods to be supplied. <b>Limit:</b> If the operation type is <code>operation = export</code> , maximum 500,000 per GTIN
certificate	No	String	Global Unique Identifier (GUID) of the electronic

			veterinary accompanying document in SIS "AITS"
name	No	String	Item name
product_cost	No	String, Integer	Unit price (including VAT, if the goods are subject to VAT). <i>See note 2 below.</i>
product_tax	No	String, Integer	Total unit VAT. <i>See note 2 below.</i>
certificate_document_data	No	Array of Objects	Array of information on documents confirming compliance of goods with quality and/or safety requirements
certificate_type	Yes	String	Type of document confirming compliance (authorization document). Enumeration value code <code>certificate_types</code> retrieved from <code>/directories/{:code}</code> method (see 3.8.3; the list of values is given in Table 4.3.6, acceptable values are given in Annex F)
certificate_number	Yes	String	Number of the document confirming compliance
certificate_date	Yes	Date	Date of the document confirming compliance. <b>Format:</b> YYYY-MM-DD

**Request body example (fragment)**

*Option 1. Export registration*

```
{
  "group": "milk",
  "shipping_doc": "tnttn",
  "operation": "export",
  "nomer_tn": "ГЖ0446918",
  "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": "RU-UEGCB000000000000000000037-0010",
    "gtin": {
        "04648349498834": {
            "count": 24
        },
        "04690228010347": {
            "count": 48
        },
        ...
    },
    "comment": "Принято частично по фактическому количеству"
}

```

**Table 3.7.6.2 HTTP response parameters, Code 200**

Key	Data type	Description
	Object	Supply details. Ship class object (see <a href="#">4.1.7</a> )
<b>Response body example (fragment)</b>		
<b>Option 1. Export registration</b>		
<pre> {     "created_at": "2021-05-30 10:31:45",     "name": "ГЖ0446918",     "comment": "Поставка по заказу N 9848 от 14.05.2021",     "price": {         "price": "23456.78",         "currency": "rub643"     },     "contract": false,     "operation_date": "2021-05-28",     "shipping_doc": {         "name": "ТН - товарная накладная",         "code": "tnttn"     },     "code": "498662",     "user_from": {         "name": "С000 \"Свиток\"",         "address": "г. Минск, ул. Тимирязева, д.2, ком. 16",     } } </pre>		

```
    "unp": 190704463,
    "gln": "4811612900000",
    "country": "112",
    "is_rules_agree": true
  },
  "user_to": {
    "name": "AO \"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": {
    "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"
    },
    ...
  },
  "document_id": "BY-00000000000000498662",
  "shipment_status": {
    "code": null,
    "message": ""
  }
}
```

**Option 2. Import registration**

```
{
  "created_at": "2022-11-09 10:09:17+03",
  "name": "0123456",
  "comment": null,
  "price": {
```

```
        "price": null,
        "currency": null
    },
    "contract": false,
    "operation_date": "2022-11-09",
    "shipping_doc": {
        "code": "tnttn",
        "name": "ТН – товарная накладная"
    },
    "code": 499549,
    "user_from": {
        "name": " AO \"Тандер\"",
        "unp": "2310031475",
        "country": "643",
        "address": "350002 г.Краснодар, ул.Леваневского, д.185",
        "gln": "",
        "agent": 18774,
        "is_verified": false
    },
    "user_to": {
        "name": "С000 \"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": 25
        },
        "04690228010347": {
            "count": 50
        },
        ...
    },
    "document_id": "RU-UEGCB000000000000000000037-0010",
    "accepted": {
        "04648349498834": {
            "name": "сгущенка",
            "count": 24,
            "certificate": "153d164d-9808-4fdz-b721-d92962t46236"
        },
        "04690228010347": {
            "name": "молоко питьевое цельное пастеризованное",
            "count": 48,
            "certificate": "a785hca2-3934-8b03-a7e1-62ba0efcsee"
        },
        ...
    },
    "shipment_status": {
        "code": "ACCEPTED",
        "message": "Принят"
```

```
}
}
```

**Notes:**

1. This method is used only for sending information on supply (shipment) of goods from the group "Dairy products" to the Russian Federation in the context of volume and grade accounting, as well as for import from the Russian Federation.

2. Additional information on unit price (attribute "product\_cost") and total unit VAT (attribute "product\_tax") apply in case of shipment to the Russian Federation and is sent:

2.1. On voluntary basis, including within contractual relations between the exporter (seller, shipper) and importer (buyer, recipient).

2.2. In Russian rubles (RUB).

2.3. In one of the following formats:

*Format 1.* In rubles as a number with two decimal places (digits) represented 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 Method "Dairy product supply list"**

Request type: POST

Request string: /ships/gtinList

Parameters:

**Table 3.7.7.1 HTTP request parameters**

Key	Required	Data type	Description
date_from	Yes	Date	Interval start date. <i>Format:</i> YYYY-MM-DD
date_to	Yes	Date	Interval end date. <i>Format:</i> YYYY-MM-DD
operation	Yes	String	Operation type: export: export registration; import: import registration
page	Yes	Integer	Supply list page number
Request body example			
<pre>{   "date_from": "2021-05-28",   "date_to": "2021-05-28",   "operation": "export",   "page": 1 }</pre>			

**Table 3.7.7.2 HTTP response parameters, Code 200**

Key	Data type	Description
count	Integer	Total number of supplies in the sample with current selections (filters)
ships_list	Array of Objects	Array containing information about supplies of dairy products. Ship class

	objects (see <a href="#">4.1.7</a> )
Response body example (fragment)	
<pre> {   "count": 6,   "ships_list": [     {       "created_at": "2021-05-28 16:07:14+03",       "name": "НД0294125",       "comment": "Поставка по заказу N 9848 от 14.05.2021",       "price": {         "price": "2050.75",         "currency": "rub643"       },       "contract": false,       "operation_date": "2021-05-28",       "shipping_doc": {         "name": "ТН - товарная накладная",         "code": "tnttn"       },       "code": 498655,       "user_from": {         "name": "С000 \"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": {         "04810005000853": {           "count": 8,           "certificate": "e5df5f0e-724e-4169-b5c2-426983a5eafe",           "name": "Молоко",           "product_cost": "2050.75",           "product_tax": "205.08"         }       },       "document_id": "BY-00000000000000498655",       "shipment_status": {         "code": "WAIT_ACCEPTANCE",         "message": "Ожидает приемку"       }     },     ...   ] } </pre>	

```
]
}
```

### 3.7.8 Method "List of supplies from the EAEU"

Request type: POST  
Request string: /ships/importList  
Parameters:

**Table 3.7.8.1 HTTP request parameters**

Key	Required	Data type	Description
date_from	Yes	Date	Interval start date. <b>Format:</b> YYYY-MM-DD
date_to	Yes	Date	Interval end date. <b>Format:</b> YYYY-MM-DD
page	Yes	Integer	Supply list page number
country	No	String	Digital code of the EAEU member state from which the cross-border shipment (supply) was received. Enumeration value code country obtained by /directories/{:code} method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.1</a> )
Request body example			
<pre>{   "date_from": "2021-11-01",   "date_to": "2021-11-30",   "page": 1,   "country": "643" }</pre>			

**Table 3.7.8.2 HTTP response parameters, Code 200**

Key	Data type	Description
count	Integer	Total number of supplies in the sample with current selections (filters)
shipments_list	Array of Objects	Array containing information about supplies. Import class objects (see <a href="#">4.1.10</a> )
Response body example (fragment)		
<pre>{   "count": 6,   "shipments_list": [     {       "document_id": "1ed99575-8f9d-4470-bfe4-e39be9ab5299",       "exporter_unp": "9715225506",       "exporter_name": "000 ТрейдЛогСервис",       "group": "shoes",       "type": "manual",       "created_at": "2022-11-23 09:06:04+03",       "user_to": {         "name": "Предприятие РБ для тестирования",         "address": "г. Минск, ул. Тимирязева, д.2, ком. 16",         "unp": "190704463", </pre>		

```
        "gln": "4811612900000",
        "country": "112",
        "is_rules_agree": true
    },
    "document_date": "2022-11-23",
    "document_number": "L000980S",
    "status": {
        "code": "10",
        "message": "Обработан"
    }
},
{
    "document_id": "f1s13r1z-9sh4-9dRs-gd1x-gc3x69ay3215",
    "exporter_unp": "9715225506",
    "exporter_name": "000 ТрейдЛогСервис",
    "group": "shoes",
    "type": "export",
    "created_at": "2022-11-23 09:14:05+03",
    "sender_country_code": "643",
    "user_to": {
        "name": "Предприятие РБ для тестирования",
        "address": "г. Минск, ул. Тимирязева, д.2, ком. 16",
        "unp": "190704463",
        "gln": "4811612900000",
        "country": "112",
        "is_rules_agree": true
    },
    "document_processing_date": "2022-11-23",
    "status": {
        "code": "10",
        "message": "Обработан"
    }
},
{
    "document_id": "14979997-d854-4215-820f-65862102479f",
    "exporter_unp": "9715225506",
    "exporter_name": "000 ТрейдЛогСервис",
    "group": "shoes",
    "type": "ship",
    "created_at": "2022-11-23 24:38:13+03",
    "sender_country_code": "643",
    "user_to": {
        "name": "Предприятие РБ для тестирования",
        "address": "г. Минск, ул. Тимирязева, д.2, ком. 16",
        "unp": "190704463",
        "gln": "4811612900000",
        "country": "112",
        "is_rules_agree": true
    },
    "document_processing_date": "2022-11-23",
    "document_number": "5",
    "document_date": "2022-11-18",
    "exporter_taxpayer_id": "9715225506",
    "info": {
        "received": 28,
        "not_received": 2
    }
},
```

```

        "status": {
            "code": 30,
            "message": "Оприходован"
        },
        ...
    ]
}

```

### 3.7.9 Method "Acceptance of supply from the EAEU"

Request type: POST  
Request string: /ships/addImport  
Parameters:

Table 3.7.9.1 HTTP request parameters

Key	Required	Data type	Description
group	Yes	String	Product group (the list of values is given in Table 4.2.1.2)
document_id	Yes	String	Unique identifier of the document with supply type type = ship or type = export by which the acceptance is performed obtained by method /ships/importList (see 3.7.8). <b>Note:</b> specified when the following conditions are simultaneously met: 1. Information on the supply is sent by the exporter and received in SIS "Electronic Mark"; 2 The current status of the supply document is "Processed" (status.code = 10); 3. Deferred or partial circulation of LCs is planned, including in multiple steps.
document_date	Yes/No	Date	Date of the document confirming the purchase of the item. <b>Format:</b> YYYY-MM-DD. <b>Note:</b> Required if the supply document for which acceptance is performed has type type = export
document_number	Yes/No	String	Number of the document confirming the purchase of the goods. <b>Note:</b> Required if the supply document for which acceptance is performed has type type = export
exporter_unp	Yes/No	String	Taxpayer's Number or other equivalent identifier of the seller organization (supplier, shipper or exporter) from a EAEU member state. <b>Note:</b> Required if the supply document for which acceptance is performed has type type = export
exporter_name	Yes/No	String	Name of the seller organization (supplier,

			shipper or exporter) from a EAEU member state. <b>Note:</b> Required if the supply document for which acceptance is performed has type <code>type = export</code>
labels	Yes	Array of Strings	Array of identification codes accepted for accounting, sent during cross-border shipment (supply). <b>Note:</b> Only the identification codes actually accepted for accounting from the supply specified by <code>document_id</code> are listed.
<b>Request body example (fragment)</b>			
<b>Option 1.</b> Trust acceptance based on the information available to the importer			
<pre>{   "group": "shoes",   "document_date": "2022-11-23",   "document_number": "L000980S",   "exporter_unp": "9715225506",   "exporter_name": "000 ТрейдЛогСервис",   "labels": [     "010468036426613521AWO-gnMkmL.nQ",     "010463009450755821EngFpOSW2ln5U",     "0104630094507114214PDZkODZ&gt;=wyl",     ...     "010463009450740421eC!uBoXfj*ZWF"   ] }</pre>			
<b>Option 2.</b> Trust acceptance based on a cross-border shipment document ( <code>type = ship</code> ) or a supply register ( <code>type = export</code> )			
<pre>{   "group": "shoes",   "document_id": "f1s13r1z-9sh4-9dRs-gd1x-gc3x69ay3215",   "document_date": "2022-11-20",   "document_number": "0001234",   "exporter_unp": "9715225506",   "exporter_name": "000 ТрейдЛогСервис",   "labels": [     "01046300945071142154Q0mQgbbhy2f",     "0104630094507114215pONkFIgI3JJM",     "0104630094507114215fHGdolEgjkvR",     ...     "0104630094507114215NvFUN+,%upsh"   ] }</pre>			

**Table 3.7.9.2** HTTP response parameters, Code 200

Key	Data type	Description
	Object	Supply details. Import class object (see <a href="#">4.1.10</a> )
<b>Response body example</b>		
<b>Option 1.</b> Trust acceptance based on the information available to the importer		
<pre>{   "document_id": "1ed99575-8f9d-4470-bfe4-e39be9ab5299",   "exporter_unp": "9715225506",   "exporter_name": "000 ТрейдЛогСервис", }</pre>		

```

"group": "shoes",
"type": "manual",
"created_at": "2022-11-23 09:06:04+03",
"user_to": {
  "name": "Предприятие РБ для тестирования",
  "address": "г. Минск, ул. Тимирязева, д.2, ком. 16",
  "unp": "190704463",
  "gln": "4811612900000",
  "country": "112",
  "is_rules_agree": true
},
"document_date": "2022-11-23",
"document_number": "L000980S",
"status": {
  "code": "10",
  "message": "Обработан"
}
}

```

**Option 2.** Trust acceptance based on a cross-border shipment document (type = ship) or a supply register (type = export)

```

{
  "document_id": "f1s13r1z-9sh4-9dRs-gd1x-gc3x69ay3215",
  "exporter_unp": "9715225506",
  "exporter_name": "000 ТрейдлогСервис",
  "group": "shoes",
  "type": "export",
  "created_at": "2022-11-23 09:14:05+03",
  "sender_country_code": "643",
  "user_to": {
    "name": "Предприятие РБ для тестирования",
    "address": "г. Минск, ул. Тимирязева, д.2, ком. 16",
    "unp": "190704463",
    "gln": "4811612900000",
    "country": "112",
    "is_rules_agree": true
  },
  "document_processing_date": "2022-11-23",
  "status": {
    "code": "30",
    "message": "Оприходован"
  }
}

```

### 3.7.10 Method "Information about supply from the EAEU"

Request type: POST  
Request string: /ships/import  
Parameters:

**Table 3.7.10.1** HTTP request parameters

Key	Required	Data type	Description
document_id	Yes	String	UID, a unique supply identifier retrieved from /ships/importList (see <a href="#">3.7.8</a> ) or /ships/addImport method (see <a href="#">3.7.9</a> )

details	No	Boolean	Flag for displaying detailed information about the processing result for each identification code: False: no details displayed; True: details are displayed. <b>Note:</b> By default, details = false. Available for supplies with current status "Capitalized" (status = 30)
<b>Request body example</b>			
<i>Option 1.</i> Default			
<pre>{   "document_id": "f1s13r1z-9sh4-9dRs-gd1x-gc3x69ay3215" }</pre>			
<i>Option 2.</i> Display of the processing results 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 details. Import class object (see <a href="#">4.1.10</a> )
<b>Response body example (fragment)</b>		
<i>Option 1.</i> Default		
<pre>{   "document_id": "f1s13r1z-9sh4-9dRs-gd1x-gc3x69ay3215",   "exporter_unp": "9715225506",   "exporter_name": "000 ТрейдЛогСервис",   "group": "shoes",   "type": "export",   "created_at": "2022-11-23 09:14:05+03",   "sender_country_code": "643",   "user_to": {     "name": "Предприятие РБ для тестирования",     "address": "г. Минск, ул. Тимирязева, д.2, ком. 16",     "unp": "190704463",     "gln": "4811612900000",     "country": "112",     "is_rules_agree": true   },   "document_processing_date": "2022-11-23",   "status": {     "code": "30",     "message": "Оприходован"   },   "labels": [     "01046300945071142154Q0mQgbbhy2f",     "0104630094507114215pONkFIgI3JJM",     "0104630094507114215fHGd0lEgjkvR",     ...     "0104630094507114215NvFUN+,%upsh"   ] }</pre>		

**Option 2.** Display of the processing results  
for each identification code (`details = true`)

```
{
  "document_id": "14979997-d854-4215-820f-65862102479f",
  "exporter_unp": "9715225506",
  "exporter_name": "000 ТрейдЛогСервис",
  "group": "shoes",
  "type": "ship",
  "created_at": "2022-11-23 24:38:13+03",
  "sender_country_code": "643",
  "user_to": {
    "name": "Предприятие РБ для тестирования",
    "address": "Г. Минск, ул. Тимирязева, д.2, ком. 16",
    "unp": "190704463",
    "gln": "4811612900000",
    "country": "112",
    "is_rules_agree": true
  },
  "document_processing_date": "2022-11-23",
  "document_number": "5",
  "document_date": "2022-11-18",
  "info": {
    "received": 22,
    "not_received": 8
  },
  "status": {
    "code": "30",
    "message": "Оприходован"
  },
  "labels": {
    {
      "label": "010463004706197721HkR,&b16+_m38",
      "status": 1
    },
    {
      "label": "010469024497344221<HP9iSyVp47Jk",
      "status": 0
    },
    {
      "label": "010463009450755821EngFpOSW2ln5U",
      "status": 1
    },
    ...
    {
      "label": "010469024497347321yS&Awke)RpI20",
      "status": 0
    }
  ]
}
```

**3.7.11 Method "Information on the supply of dairy products"**

Request type: POST  
Request string: /ships/getGtin  
Parameters:

**Table 3.7.11.1 HTTP request parameters**

Key	Required	Data type	Description
<b>Option 1.</b> For export of dairy products (operation = export)			
code	Yes	String	Supply number. Supply ID retrieved from /ships/gtin (see <a href="#">3.7.6</a> ) or /ships/gtinList method (see <a href="#">3.7.7</a> )
<b>Option 2.</b> For the supply of dairy products by import (operation = import)			
document_id	Yes	String	Unique identifier of the document retrieved from /ships/gtinList method (see <a href="#">3.7.7</a> ).
<b>Request body example</b>			
<b>Option 1.</b> For export of dairy products (operation = export)			
<pre>{   "code": "499524" }</pre>			
<b>Option 2.</b> For the supply of dairy products by import (operation = import)			
<pre>{   "document_id": "RU-UEGCB0000000000000000000037-0010" }</pre>			

**Table 3.7.11.2 HTTP response parameters, Code 200**

Key	Data type	Description
	Object	Information on the supply of dairy products. Ship class object (see <a href="#">4.1.7</a> )
<b>Response body example (fragment)</b>		
<b>Option 1.</b> For export of dairy products (operation = export)		
<pre>{   "created_at": "2022-10-10 14:55:38+03",   "name": "004-4811612900000-0000000000000000123456789",   "comment": "",   "price": {     "price": "123456.78",     "currency": "rub643"   },   "contract": false,   "operation_date": "2022-10-10",   "shipping_doc": {     "code": "etn",     "name": "ЭТН – электронная товарная накладная"   },   "code": 499524,   "user_from": {     "name": "C000 \"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", </pre>		

```

        "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": "BY-00000000000000499524",
    "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.** For 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": null
    },
    "contract": false,
    "operation_date": "2022-11-09",
    "shipping_doc": {

```

```

        "code": "tnttn",
        "name": "ТН – товарная накладная"
    },
    "code": 499549,
    "user_from": {
        "name": " AO \"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": 25
        },
        "04690228010347": {
            "count": 50
        },
        ...
    },
    "document_id": "RU-UEGCB000000000000000000000037-0010",
    "shipment_status": {
        "code": "WAIT_ACCEPTANCE",
        "message": "Ожидает приемку"
    }
}

```

### 3.7.12 Method "Full acceptance of codes"

#### 3.7.12.1. Representation A

Request type: POST  
Request string: /ships/takeAll  
Parameters:

**Table 3.7.12.1** HTTP request parameters

Key	Required	Data type	Description
code	Yes	Integer	Shipment number. Shipment ID retrieved from method /ships/list (see <a href="#">3.7.5.1</a> )
<b>Request body example</b>			
{			

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

**Table 3.7.12.2** HTTP response parameters, Code 200

Key	Data type	Description
message	String	Information message
count	Integer	Total number of capitalized codes (including LCs and ACs at all nesting levels)
Response body example		
<pre>{   "message": "Успешно оприходовано",   "count": 15 }</pre>		

### 3.7.12.2. Representation B

Request type: POST  
Request string: /v3/ships/take\_all  
Parameters:

**Table 3.7.12.3** HTTP request parameters

Key	Required	Data type	Description
ship_id	Yes	Integer	Shipment number. Shipment ID retrieved from method /v3/ships/list (see <a href="#">3.7.5.2</a> )
Request body example			
<pre>{   "ship_id": 2000040045 }</pre>			

**Table 3.7.12.4** HTTP response parameters, Code 200

Key	Data type	Description
detail	String	Information message
ship_id	Integer	Number (identifier) of the shipment being accepted
Response body example		
<pre>{   "detail": "Приемка начата",   "ship_id": 2000040045 }</pre>		

### 3.7.13 "List of accepted goods supplied from the EAEU"

Request type: POST  
Request string: /ships/import  
Parameters:

**Table 3.7.10.1** HTTP request parameters

Key	Required	Data type	Description
-----	----------	-----------	-------------

document_id	Yes	String	UID, a unique supply identifier retrieved from /ships/importList (see <a href="#">3.7.8</a> ) or /ships/import method (see <a href="#">3.7.10</a> )
Request body example			
<pre>{   "document_id": "RU-U ECB000000000000000000000000157-0289" }</pre>			

**Table 3.7.13.2** HTTP response parameters, Code 200

Key	Data type	Description
document_id	String	UID, a unique supply identifier
gtin_list	Array of Objects	Array containing a list of accepted goods
gtin	String	Item GTIN
name	String	Item name
count	Integer	Number of units of accepted goods
Response body example		
<pre>{   "document_id": "RU-U ECB000000000000000000000000157-0289",   "gtin_list": [     {       "gtin": "04548515036139",       "name": "Шина пневматическая YOKOHAMA ADVAN Neova AD09 295/30R20 101W.",       "count": 4     },     {       "gtin": "04811644000730",       "name": "Шина пневматическая для легковых автомобилей",       "count": 6     }   ] }</pre>		

## 3.8 Auxiliary methods

### 3.8.1 Method "File download"

Request type: GET  
Request string: /downloads/{:filename}  
Parameters:

**Table 3.8.1.1** HTTP request parameters

Key	Required	Data type	Description
filename	Yes	String	File name. Usually specified in the same-named parameter of section "file" of the Order class object (see <a href="#">4.1.6</a> ), returned by /orders/{:id} method (see <a href="#">3.3.2.1</a> )
Request string example			
/downloads/507-sygn-253590-2020-11-19-14-01-01-4.txt			

### 3.8.2 Method "Enumeration directories list"

Request type: POST  
Request string: /directories/  
Parameters:

**Table 3.8.2.1** HTTP response parameters, Code 200

Key	Data type	Description
directories	Array of Objects	Array containing information about enumeration directories. Directory class objects (see <a href="#">4.1.8</a> )
<b>Response body example (fragment)</b>		
<pre>{   "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     },     ...   ] }</pre>		

### 3.8.3 Method "Enumeration directory elements"

Request type: POST  
Request string: /directories/{:code}  
Parameters:

**Table 3.8.3.1** HTTP request parameters

Key	Required	Data type	Description
code	Yes	String	List code obtained by /directories/ method (see <a href="#">3.8.2</a> )
<b>Request string example</b>			
/directories/currency			

**Table 3.8.3.2** HTTP response parameters, Code 200

Key	Data type	Description
	Object	Description of a directory (list, enumeration)

		and a list of its elements. Directory class object (see <a href="#">4.1.8</a> )
<b>Response body example</b>		
<pre> {   "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     }   ] } </pre>		

**Note:** The values of some enumeration directories used as parameter values of individual API methods are given in Section [4.3](#) of this document.

### 3.8.4 Method "Printing house list"

Request type: POST  
Request string: /users/typographyList  
Parameters: No input parameters

**Table 3.8.4.1** HTTP response parameters, Code 200

Key	Data type	Description
typograpy_list	Array of Objects	Array with a list of printing houses. Agent class object (see <a href="#">4.1.2</a> )
<b>Response body example (fragment)</b>		
<pre> {   "typography_list": [     {       "id": 2643,       "name": "000 \"Типография\"", </pre>		

```

        "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 Billing handling methods

#### 3.9.1 Method "Service delivery report list"

Request type: GET  
 Request string: /v3/common/getUsersBillingPdfActsList  
 ?[act\_type={:act\_type}]  
 &[page={:page}]  
 &[page\_size={:page\_size}]  
 &[ordering={:ordering}]  
 &[period\_year\_start={:period\_year\_start}]  
 &[period\_month\_start={:period\_year\_start}]  
 &[period\_year\_end={:period\_month\_end}]  
 &[period\_month\_end={:period\_month\_end}]

Parameters:

**Table 3.9.1.1** HTTP request parameters

Key	Required	Data type	Description
act_type	No	Integer	Group of services provided. Allowed values: 9: LC issuance and accounting services; 10: "User account" access services.
page	No	Integer	Service delivery report list page number
page_size	No	Integer	Number of entries per page. <i>Note:</i> Maximum 200, by default, up to 15 per page.

ordering	No	String	Data sorting conditions: id: by ascending identifier of service delivery report; - id: by descending identifier of service delivery report
period_year_start	No	Integer	Settling period interval start year
period_month_start	No	Integer	Settling period interval start month
period_year_end	No	Integer	Settling period interval end year
period_month_end	No	Integer	Settling period interval end month
<b>Request string example</b>			
/common/getUsersBillingPdfActsList?page=1&page_size=15&period_year_start=2024&period_month_start=12&period_year_end=2025&period_month_end=2&act_type=9			

**Table 3.9.1.2 HTTP response parameters, Code 200**

Key	Data type	Description
next	String	Link to the next page of the service delivery report list. <i>Note:</i> Null if none.
previous	String	Link to the previous page of the service delivery report list. <i>Note:</i> Null if none.
results	Array of Objects	Array containing information on service delivery reports. ServiceAcceptanceCertificate class objects (see <a href="#">4.1.12</a> )
<b>Response body example (fragment)</b>		
<pre>{   "next": null,   "previous": null,   "results": [     {       "id": 2906,       "user_info_id": 129,       "period_year": 2025,       "period_month": 1,       "info": {         "billing_data_gln": [           {             "nds": "8.45",             "sum": "42.25",             "url": "/api/calculations/act/print/190704463_5500_31012025.pdf",             "date": "2025-01-31",             "price": "0.0170",             "number": 5500,             "sum_nds": "50.70",             "act_name": "Услуга по предоставлению и учету кода маркировки (Код РБ идентификации товара)",             "quantity": 2485,             "label_type": "12, 20",             "one_c_code": "яв806",             "period_end_date": "2025-01-31",             "period_start_date": "2025-01-01"           }         ]       }     }   ] }</pre>		

```
    ]
    },
    "admin_billing_pdf_acts_id": 24,
    "user_info": {
        "name": "C000 \"Свиток\"",
        "address": "г. Минск, ул. Тимирязева, д.2, ком. 16",
        "unp": "190704463",
        "gln": "4811612900000",
        "country": "112",
        "is_rules_agree": true,
        "user_info_id_old": null
    },
    "base_url": "https://billing.4ek.by:80"
},
{
    "id": 1955,
    "user_info_id": 129,
    "period_year": 2025,
    "period_month": 1,
    "info": {
        "billing_data_gln": [
            {
                "nds": "2.92",
                "sum": "14.61",
                "url": "/api/calculations/act/print/190704463_4852_31012025.pdf",
                "date": "2025-01-31",
                "price": "0.0300",
                "number": 4852,
                "sum_nds": "17.53",
                "act_name": "Услуги по предоставлению кодов маркировки  
(Код стран ЕАЭС идентификации товара)",
                "quantity": 487,
                "label_type": "7",
                "one_c_code": "эе666",
                "period_end_date": "2025-01-31",
                "period_start_date": "2025-01-01"
            }
        ]
    },
    "admin_billing_pdf_acts_id": 18,
    "user_info": {
        "name": "C000 \"Свиток\"",
        "address": "г. Минск, ул. Тимирязева, д.2, ком. 16",
        "unp": "190704463",
        "gln": "4811612900000",
        "country": "112",
        "is_rules_agree": true,
        "user_info_id_old": null
    },
    "base_url": "https://billing.4ek.by:80"
},
...
]
```

## 4 OBJECTS, DIRECTORIES AND ENUMERATIONS

The description of the composition and structure of object parameters and possible values given in this section is for reference only and may be changed in the next API versions. The output of certain parameters of an object in an HTTP response is determined by the specific API method that uses the object. Typical uses of objects are given as examples in the description of the corresponding API method.

### 4.1 Object classes

#### 4.1.1 Object "User"

Class name: User

Parameters:

**Table 4.1.1** User class metadata structure

Key	Data type	Description
id	Integer	User code (identifier)
name	String	First name
lastname	String	Last name
email	String	E-mail address
status	Integer	Current status: 0: disabled; 1: active
expires_in	Date	Authentication token validity period. <i>Note:</i> Used in the authorization method /auth (see <a href="#">3.1.1</a> )
role	Object	Participant role. UserRole class object (see <a href="#">4.2.7</a> )
info_id	Integer	Subject (participant) ID
need_change_password	Boolean	Flag of the need for password change: False: not required; True: required
info	Object	Agent (participant) details. Agent class object (see <a href="#">4.1.2</a> )

#### 4.1.2 Object "Agent"

Class name: Agent

Parameters:

**Table 4.1.2** Agent class metadata structure

Key	Data type	Description
id	Integer	Agent code (identifier). <i>Note:</i> The specified attribute is named as "agent" in the "user_to" parameter of the Ship class object
name	String	Agent name
adres	String	Agent address
unp	String	Taxpayer's Number or other equivalent identifier of the agent

Key	Data type	Description
gln	String	Agent GLN
country	String, Object	Country of registration of the agent. Enumeration value code <code>country</code> obtained by <code>/directories/{:code}</code> method (see <a href="#">3.8.3</a> ; the list of values is given in Table 4.3.1)
is_rules_agree	Boolean	Flag of having read and accepted the terms and conditions of the public contract with the Operator on the provision of IM and services on provision and accounting of LC in SIS "Electronic Mark" (accession to the public contract): True: yes; False: no.
is_verified	Boolean	Flag of the circulation participant registered in SIS "Electronic Mark": True: is a circulation participant; False: not a circulation participant.
status	Object	Agent status. Flag of active (enabled) record ( <code>status.code = 1</code> ), or blocked (disabled) record ( <code>status.code = 0</code> ).
code	String	Status code
message	String	Status name

### 4.1.3 Object "Product catalog section (category)"

Class name: Category

Parameters:

**Table 4.1.3.1** Category class metadata structure

Key	Data type	Description
code	Integer	Catalog section code (identifier)
name	String	Catalog section name
group	Object	Product group details. Group class object (see <a href="#">4.2.1</a> )
inner	Array of Objects	Array of child (lower level, nested) catalog sections. <b>Category</b> class objects
catalog_params	Array of Objects	Array (set, composition) of additional parameters assigned to a catalog section, describing the product as a whole
group	String	Parameter group name
params	Array of Objects	A set of parameters and characteristics. <b>Param</b> class objects (see <a href="#">4.1.5</a> )
mark_params	Array of Objects	Array (set, composition) of additional parameters assigned to a catalog section, describing the batch of labeled items
group	String	Parameter group name
params	Array of Objects	A set of parameters and characteristics. <b>Param</b> class objects (see <a href="#">4.1.5</a> )
items	Array of Objects	Array containing descriptions of products of

Key	Data type	Description
		the specified catalog section, added by the participant (i.e., having the <code>is_my = True</code> flag). Item class objects (see <a href="#">4.1.4</a> )

The values of the directory "Catalog Sections" are given in Table [4.1.3.2](#).

**Table 4.1.3.2** List of product catalog sections

code / Code	name / Name
10002	Footwear
10081	Consumer goods
10082	Bed linen, table linen, bathroom linen and kitchen linen
10083	Clothing
10086	Tires
10087	Perfume and eau de toilette
10088	Cameras and flash lamps
10089	Dairy products
10090	Goods subject to UCM labeling
10092	Residual stock. Footwear
10093	Residual stock. Tires
10094	Packaged water
10095	Residual stock. Consumer goods
10096	Non-alcoholic beverages and juices
10097	Dietary supplements
10098	Vegetable oils and fat-and-oil products
10099	Preserves
10100	Animal feed
10101	Veterinary medicines
10102	Sturgeon roe and salmon roe
10103	Bicycles and bicycle frames
10104	Antiseptics and disinfectants
10105	Cosmetics and household cleaning products
10106	Food products (Groceries)

**Note:** Catalog section identifiers (codes) are different in the commercial and test sections. The table shows the catalog category codes used in the commercial section.

#### 4.1.4 Object "Item"

Class name: Item

Parameters:

**Table 4.1.4.1** Item class metadata structure

Key	Data type	Description
gtin	String	Item GTIN
name	String	Item name
tnved	String	EAEU CN of FEA code
<i>For items registered in the catalog</i>		
image	String	URL link to the item image

Key	Data type	Description
articul	String	Item number
code	String	Internal code (identifier) of the item details record
description	String	Item description
gtin_check	Boolean	Flag of the GTIN registration in the catalog: True: the item is registered; False: the item is pending registration.
group	String	Product group (the list of values is given in Table 4.2.1.2)
debug_message	Object	Service message
is_my	Boolean	Custom item flag: True: item added by the current participant; False: item added by another participant;
catalog	Object	Catalog section details. <b>Category</b> class object (see 4.1.3)
params	Array of Objects	Array of additional parameters describing the product. The set of parameters (additional characteristics) and conditions of their setting is determined by the belonging of the product to the corresponding section of the catalog. Param class objects (see 4.1.5) listed in the attribute "catalog_params" of the <b>Category</b> class object (see 4.1.3)
<i>For imported goods</i>		
unp	String	Taxpayer's Number or other equivalent identifier of the item card owner
brand	String	Brand, trademark
country	Object	Country of origin of the item. Enumeration value <b>country</b> obtained by <code>/directories/{:code}</code> method (see 3.8.3; the list of values is given in Table 4.3.1).
is_import	Boolean	Flag of item imported from the catalogs of national operators of the EAEU member states
operator_country	String	Digital country code of the operator of the EAEU member state that provided the description of the item. Enumeration value <b>country</b> obtained by <code>/directories/{:code}</code> method (see 3.8.3; the list of values is given in Table 4.3.1)
epass_sync	Boolean	Flag of synchronization of product information with the Electronic Product Passport Data Bank (ePASS): True: product information registered in ePASS; False: product description is not synchronized with ePASS.
<i>Information on an instance of labeled items</i>		
mark_params	Array of Objects	Array of additional parameters describing the

Key	Data type	Description
		instance of labeled items. The set of parameters (additional characteristics) and conditions of their setting is determined by the belonging of the product to the corresponding section of the catalog. Param class objects (see <a href="#">4.1.5</a> ) listed in the attribute "mark_params" of the Category class object (see <a href="#">4.1.3</a> )

The values of the directory "Items" related to the product group "Items subject to UCM labeling" (group = ukz) are given in Table [4.1.4.2](#).

**Table 4.1.4.2** List of sections of items subject to UCM labeling

code / Code	name / Name
100900000695388	Caviar; canned fish products (EAEU CN of FEA code from 1604)
100900000695389	Water (EAEU CN of FEA code from 2202)
100900000695390	Clocks (EAEU CN of FEA code from 9101, 9102, 9105)
100900000695391	TV sets (EAEU CN of FEA code from 8528)
100900000695392	Monitors (EAEU CN of FEA code from 8528)
100900000695393	Physical data carriers (EAEU CN of FEA code from 8523)
100900000695394	Mobile phones in consumer packaging, except for used (EAEU CN of FEA code from 8517 13 000 0, 8517 14 000 0)
100900000695395	Laptops, tablet computers (EAEU CN of FEA code from 8471 41 000 0)
100900000695396	Laptops, tablet computers (EAEU CN of FEA code from 8471 30 000 0)
100900000695397	Printers (EAEU CN of FEA code from 8443 32 100 9)
100900000695398	Liquids for non-combustible tobacco products (EAEU CN of FEA code from 2404, 3824 99)
100900000695399	Antifreezes (EAEU CN of FEA code from 3820 00 000 0)
100900000695400	Detergents (EAEU CN of FEA code from 3402)
100900000695401	Engine oil (EAEU CN of FEA code from 2710 19, 3403)
100900000695402	Low-alcoholic beverages (EAEU CN of FEA code from 2204-2206 00, 2208)
100900000695403	Beer (EAEU CN of FEA code from 2203 00)
100900000695404	Instant coffee; instant tea (EAEU CN of FEA code from 2101)
100900000695405	Vegetable oil (EAEU CN of FEA code from 1507-1517)
100900000695406	Mate (Paraguayan tea) (EAEU CN of FEA code from 0903 00 000 0)
100900000695407	Tea (EAEU CN of FEA code from 0902)
100900000695408	Coffee (EAEU CN of FEA code from 0901)
100900000695409	Juices (EAEU CN of FEA code from 2009)

**Note:** Identifiers (codes) of items subject to UCM labeling are different in the commercial and test sections. The table shows the codes of items subject to UCM labeling used in the commercial section.

### 4.1.5 Object "Parameter"

Class name: Param

Parameters:

**Table 4.1.5** Param class metadata structure

Key	Data type	Description
code	Integer	Parameter code (identifier)
name	String	Parameter name
type	String	Element type: text: text field; select: drop-down list; directory: directory; checkbox: Boolean, True or False, date: date in the format YYYY-MM-DD; dynamic: parameter with dynamic default value
slug	String	Key (string designation) of the parameter
value	String, Integer, Decimal, Date, Boolean, Object	Parameter value, depends on the type
code	String	Name of the enumeration directory (list code) used for the element with type type = directory
list	Array	List of allowed values
display_value	String	Displayed value, representation
units	Object	Unit of measurement
is_required	Boolean	Flag of a required value: True: null values are not allowed; False: value can be null.
is_editable	Boolean	Flag whether values can be edited by method item/edit: True: allowed; False: not allowed.
is_show_catalog	Boolean	Visibility flag for (output, displaying) the parameter in the product information in the catalog: True: visible; False: invisible.

### 4.1.6 Object "Order"

Class name: Order

Parameters:

**Table 4.1.6** Order class metadata structure

Key	Data type	Description
id	String, Integer	Order number (identifier)
created_at	Date	Order creation date and time
completed_at	Date	Order execution date and time
updated_at	Date	Date of last update of information

Key	Data type	Description
comment	String, Object	Order comment
type	Object	Type of codes ordered. LabelType class object (see <a href="#">4.2.2</a> )
status	Object	Current order status. OrderStatus class object (see <a href="#">4.2.4</a> )
user	Object	User created the order. User class object (see <a href="#">4.1.1</a> )
<i>Представление А</i>		
parent_order	Integer	Number (identifier) of the order based on which the current order was created
gtin	String	Item GTIN
count	Integer	Total number of codes ordered
file	Object	File for downloading codes. File class object (see <a href="#">4.1.13</a> )
labels	Array of Strings	Array of ordered codes
<i>Представление Б</i>		
gtin	String, Object	Item GTIN or description (Item class object, see <a href="#">4.1.4</a> )
item	Object	Description of the item with the specified GTIN. Item class object (see <a href="#">4.1.4</a> )
group	Object	Product group details. Group class object (see <a href="#">4.2.1</a> )
count	Object	Number of codes ordered:
all	Integer	- total;
success	Integer	- successful;
failed	Integer	- incorrect
files	Object	List of file names with ordered codes
{:file_type}	String	Element key name corresponds to the type of information returned: labels: LC; aggregates: AC.
filename	String	Name of the file containing the ordered codes
downloads	Integer	Number of downloads
user_id	Integer	Code (identifier) of the user that created the order
user_info_id	Integer	Code (identifier) of the circulation participant that created the order
user_info	Object	Details of the circulation participant that created the order. Agent class object (see <a href="#">4.1.2</a> )

#### 4.1.7 Object "Shipment"

Class name: Ship

Parameters:

**Table 4.1.7** Ship class metadata structure

Key	Data type	Description
name	String	Accompanying document number
created_at	Date	Shipment creation date and time

Key	Data type	Description
comment	String	Shipment comment
count	Integer	Total number of units of labeled goods according to the underlying document
taked_at	Date	Date and time of the code acceptance (information submission) operation by the recipient
items	Array of Objects	Array containing the descriptions of the items shipped. Item class objects (see <a href="#">4.1.4</a> )
labels	Array of Objects	Array of information on types of codes shipped and their quantity
id	Integer	Code (identifier) of the code type
name	String	Code type name
count	Integer	Number of codes of the specified type shipped
operation_date	Date	Date of shipment (usually corresponds to the date of the accompanying document)
shipping_doc	Object	Type of accompanying document. Enumeration value code shippingDocs obtained by /directories/{:code} method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.3</a> )
status	Object	Current shipment status. ShipStatus class object (see <a href="#">4.2.6</a> )
country	Object	Country to which the shipment is made. Enumeration value country obtained by /directories/{:code} method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.1</a> ).
eas_products	Object	Additional information about the goods to be supplied. <b>Note:</b> Used in case of transfer of shipment information in cross-border trade between the EAEU member states.
{:gtin}	String	The element key name corresponds to the GTIN value of the item. The element value contains additional parameters of shipment by the specified GTIN
product_cost	String	Unit price (including VAT, if the goods are subject to VAT)
product_tax	String	Total unit VAT
certificate_document_data	Array of Objects	Array of information on documents confirming compliance of goods with quality and/or safety requirements or registration of goods in accordance with the established procedure
certificate_type	String	Type of document confirming compliance (authorization document). Enumeration value code certificate_types obtained by /directories/{:code} method (see

Key	Data type	Description
		<a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.6</a> )
certificate_number	String	Number of the document confirming compliance
certificate_date	Date	Date of the document confirming compliance
<i>Representation A</i>		
code	Integer	Shipment number (identifier)
real_count	Integer	Total number of shipped codes (including LCs and ACs at all nesting levels)
agg_count	Integer	Total number of ACs shipped
received_count	Integer	Total number of capitalized codes (including LCs and ACs at all nesting levels)
user_from	Object	Agent (shipper) details. Agent class object (see <a href="#">4.1.2</a> )
user_to	Object	Agent (recipient) details. Agent class object (see <a href="#">4.1.2</a> )
contract	Object	Contract (agreement) Contract class object. If not available, False
price	Object	Estimated shipment value
price	Decimal	Total cost of labeled goods according to the underlying document
currency	String	Shipment currency Enumeration value code currency obtained by /directories/{:code} method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.2</a> )
operation	String	Type of supply registration operation (export or import). <b>Note:</b> Used for information on supplies of goods of the group "Dairy products" (group = milk).
document_id	String	Unique identifier of the supply document. <b>Note:</b> Used for information on supplies of goods of the group "Dairy products" (group = milk).
gtin	Object	Parameters of volume and grade accounting of supplied goods in terms of shipped GTIN and quantity. <b>Note:</b> Used for information on supplies of goods of the group "Dairy products" (group = milk).
{:gtin}	String	The element key name corresponds to the GTIN value of the item. The element value contains supply parameters by the specified GTIN
count	Integer	Quantity of goods to be supplied
certificate	String	Global Unique Identifier (GUID) of the

Key	Data type	Description
		electronic veterinary accompanying document in SIS "AITS"
name	String	Item name
product_cost	String	Unit price (including VAT, if the goods are subject to VAT)
product_tax	String	Total unit VAT
certificate_document_data	Array of Objects	Array of information on documents confirming compliance of goods with quality and/or safety requirements
certificate_type	String	Type of document confirming compliance (authorization document). Enumeration value code <code>certificate_types</code> obtained by <code>/directories/{:code}</code> method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.6</a> )
certificate_number	String	Number of the document confirming compliance
certificate_date	Date	Date of the document confirming compliance
accepted	Object	Parameters of volume and grade accounting of supplied goods in terms of accepted GTIN and quantity. <b>Note:</b> Used for information on supplies of goods of the group "Dairy products".
<code>{:gtin}</code>	String	The element key name corresponds to the GTIN value of the item. The element value contains supply parameters by the specified GTIN
count	Integer	Quantity of accepted goods
certificate	String	Global Unique Identifier (GUID) of the electronic veterinary accompanying document
name	String	Item name
shipment_status	Object	Current status of the dairy product supply document. <b>Note:</b> Displayed and updated upon the receipt of information on the processing of the supply document in the information system of the EAEU member state
code	String	Status designation
message	String	Status description
debug_message	String	Error description (if present)
<i>Representation B</i>		
id	Integer	Shipment number (identifier)
real_count	Integer	Total number of LCs shipped
aggregate_count	Integer	Total number of top level ACs shipped
all_aggregate_count	Integer	Total cumulative number of shipped ACs at all nesting levels
received_count	Object	Total number of capitalized codes,

Key	Data type	Description
		including:
labels	Integer	- number of LCs accepted
aggregates	Integer	Cumulative number of accepted ACs at all nesting levels
agent_id	Integer	ID of the recipient counterparty to which the codes are shipped
user_info_from	Object	Agent (shipper) details. Agent class object (see <a href="#">4.1.2</a> )
user_info_to, agent_to	Object	Agent (recipient) details. Agent class object (see <a href="#">4.1.2</a> )
contract_id	Integer	ID (identifier) of the agreement (contract). Null if none
price	Decimal	Total cost of labeled goods according to the underlying document
nds_sum	Decimal	VAT amount
currency	String	Shipment currency Enumeration value code currency obtained by /directories/{:code} method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.2</a> )
files	Object	List of file names with shipped codes
{:file_type}	String	Element key name corresponds to the type of information returned: labels_sent: sent LCs; labels_taken: received LCs; labels_error: incorrect codes (LCs and/or ACs); aggregates_sent: sent ACs; aggregates_taken: received ACs; destroy_aggregates: ACs destroyed after shipment; destroy_aggregates_after_ship: ACs destroyed after acceptance. Key value contains the file name
canceled_at	Date	Date and time of shipment cancellation
cancel_comment	String	Comment on the cancel shipment operation
cancel_by_admin	Boolean	Flag of cancelled shipment: False: shipment cancelled by recipient; True: shipment cancelled by Operator.

#### 4.1.8 Object "Directory (list, enumeration)"

Class name: Directory

Parameters:

**Table 4.1.8** Directory class metadata structure

Key	Data type	Description
code	String	Key (identifier) of a directory (list, enumeration)

Key	Data type	Description
name	String	Name of a directory (list, enumeration)
comment	String	Description of a directory (list, enumeration)
inner	Array of Objects	Array of elements of a directory (list, enumeration)
code	String	Element code (identifier)
name	String	Element value
comment	String	Note, comment

#### 4.1.9 Object "Inventory"

Class name: Inventory

Parameters:

**Table 4.1.9** Inventory class metadata structure

Key	Data type	Description
group	String	Product group (the list of values is given in Table <a href="#">4.2.1.2</a> )
name	String	Inventory number
date	Date	Inventory date
quantity	Integer	Total number of the residual stock based on inventory
id	String	Inventory ID

#### 4.1.10 Object "Import"

Class name: Import

Parameters:

**Table 4.1.10** Import class metadata structure

Key	Data type	Description
document_id	String	Unique identifier of the supply (UID)
exporter_unp	String	Taxpayer's Number or other equivalent identifier of the seller organization (supplier, shipper or exporter) from a EAEU member state
exporter_name	String	Name of the seller organization (supplier, shipper or exporter) from a EAEU member state
group	String	Product group (the list of values is given in Table <a href="#">4.2.1.2</a> )
type	String	Supply type: <b>export</b> : sent by the Russian Federation operator through the supply register (temporary scheme); <b>manual</b> : added by the turnover participant themselves (trust acceptance); <b>ship</b> : provided by the exporter in a cross-border shipment document
created_at	Date	Date and time of supply registration in SIS

Key	Data type	Description
		"Electronic Mark"
sender_country_code	String	Digital code of the EAEU member state from which the cross-border shipment (supply) was received. Enumeration value code country obtained by /directories/{:code} method (see <a href="#">3.8.3</a> ; the list of values is given in Table <a href="#">4.3.1</a> )
user_to	Object	Agent (recipient) details. Agent class object (see <a href="#">4.1.2</a> )
document_date	Date	Date of the document confirming the purchase of the item
document_number	String	Number of the document confirming the purchase of the goods.
document_processing_date	Date	Date of provision of information by the operator of the EAEU member state from which the cross-border shipment (supply) was received.
info	Object	Supply processing results
received	Integer	Total number of accepted (capitalized) identification codes
not_received	Integer	Total number of non-accepted (rejected) identification codes
status	Object	Current supply processing status. ImportStatus class object (see <a href="#">4.2.9</a> )
code	String	Status code
message	String	Status name
labels	Array of Strings or Array of Objects	Array of identification codes, sent during cross-border shipment (supply).
label	String	Identification code
status	Integer	Identification code processing result: 0: non-accepted (rejected); 1: accepted (capitalized)

#### 4.1.11 Object "Aggregate"

Class name: Aggregate

Parameters:

**Table 4.1.11** Aggregate class metadata structure

Key	Data type	Description
aggregates_count	Integer	Number of lower level (child) ACs nested in the aggregate
labels_count	Integer	Number of LCs nested in the aggregate
status	String	Current aggregate status
level	Integer	Aggregate nesting level
user_info	String	Name of the current code owner
high_aggregates	Array of Objects	Array element contains a description of the upper level aggregate in which the source

Key	Data type	Description
		aggregate is included (belongs to)
uuid	String	Upper level AC (SSCC or custom)
level	Integer	Upper aggregate nesting level
aggregate_inside	Object	Nested (child) aggregates
{:uuid}	Object	The element key name corresponds to the value of the nested AC, the element value contains the description and composition of the child aggregate. <b>Aggregate</b> class object
labels_inside	Array of Strings	Array (list) of LCs nested in the aggregate

#### 4.1.12 Object "Service delivery report"

Class name: ServiceAcceptanceCertificate

Parameters:

**Table 4.1.12** ServiceAcceptanceCertificate class metadata structure

Key	Data type	Description
id	Integer	ID (identifier) of the service delivery report
user_info_id	Integer	ID (identifier) of the circulation participant to which the services were provided
period_year	Integer	Settling period year
period_month	Integer	Settling period month
info	Object	Information on the volume and cost of services provided
billing_data_gln	Array of Objects	List of services provided
date	Date	Date of services provided
number	Integer	Number of the service delivery report
period_start_date	Integer	Settling period start date
period_end_date	Integer	Settling period end date
act_name	String	Service name
one_c_code	String	Service code
label_type	String	Code type. Code ID retrieved from /labels/types method (see 3.4.3; the list of values is given in Table 4.2.2.2). <b>Note:</b> Different types of LCs ordered and accounted for within the service provided can be listed separated by commas
quantity	Integer	Amount of services provided (LCs ordered and accounted for)
price	String	Unit price (tariff)
sum	String	Amount ex. VAT, rubles, kopecks
nds	String	VAT amount, rubles, kopecks
sum_nds	String	Amount incl. VAT, rubles, kopecks
url	String	Full name of the PDF file with download URL
admin_billing_pdf_acts_id	Integer	Service identifier of the document
user_info	Object	Information on the circulation participant to which the services were provided <b>Agent</b> class object (see 4.1.2)
base_url	String	Service URL address

### 4.1.13 Object "File"

Class name: File

Parameters:

**Table 4.1.13** File class metadata structure

Key	Data type	Description
filename	String	Full file name with download URL
downloads	Integer	Number of downloads

## 4.2 System directories

### 4.2.1 Directory "Product groups"

Class name: Group

Parameters:

**Table 4.2.1** Group class metadata structure

Key	Data type	Description
code	String	Product group key (identifier)
name	String	Product group name
articul_required	Boolean	Flag of the requirement to fill in the "Article" field for the items of this group: False: not required; True: required
gtin_required	Boolean	Flag of the requirement to fill in the "GTIN" field for the items of this group: False: not required; True: required
is_manually_add	Boolean	Flag of possibility to add new products of this catalog group, method /items/add is used: False: not allowed; True: allowed
is_gtin_add	Boolean	Flag of possibility to add new products of this catalog group, method /items/addByGtin is used: False: not allowed; True: allowed
is_show_gtin	Boolean	Visibility flag for (output, displaying) the GTIN in the product information in the catalog: True: visible; False: invisible.
is_show_articul	Boolean	Visibility flag for (output, displaying) the item number in the product information in the catalog: True: visible; False: invisible.

Values of the directory "Product groups" are given in Table [4.2.1.2](#).

**Table 4.2.1.2** Product group list

code / Code	name / Name
animal_feed	Animal feed
antiseptics	Antiseptics and disinfectants
bike	Bicycles and bicycle frames
caviar	Sturgeon roe and salmon roe
clothes	Consumer goods
cosmetics	Cosmetics and household cleaning products
dietary_supplements	Dietary supplements
grocery	Food products (Groceries)
juice	Non-alcoholic beverages and juices
milk	Dairy products
oil_fat	Vegetable oils and fat-and-oil products
perfum	Perfume and eau de toilette
preserves	Preserves
photo	Cameras and flash lamps
shoes	Footwear
tires	Tires
ukz	Goods subject to UCM labeling
veterinary_medicines	Veterinary medicines
water	Packaged water
leftovers_clothes	Residual stock. Consumer goods
leftovers_shoes	Residual stock. Footwear
leftovers_tires	Residual stock. Tires

#### 4.2.2 Directory "Label types"

Class name: LabelType

Parameters:

**Table 4.2.2.1** LabelType class metadata structure

Key	Data type	Description
id	Integer	Code (identifier) of the label type
name	String	Label type name
method	String	Method of code generation of the specified type
is_bso	Boolean	Flag of a document form or a document with a certain level of protection (secure document), including those with an applied identification means: False: not a secure document; True: is a secure document
digits_count	Integer	Code length or number of digits of the number of a document with a certain level of protection of the specified type
letters_count	Integer	Number of characters of the series of a document with a certain level of protection
is_autotake	Boolean	Flag of the possibility of automatic capitalization of the code when it is transferred to another owner during shipment

Key	Data type	Description
		or sale: False: not used; True: used

Some values of the directory "Label types" are given in Table [4.2.2.2](#).

**Table 4.2.2.2** Label type list

id / Code	name / Name
6	Aggregation code (SSCC)
7	EAEU product identification code
12	Belarusian dairy product identification code
13	Aggregation code (import)
16	EAEU product identification code (photo products)
17	Unified control mark (18*18)
18	Sticker label with an identification means (18*18)
19	Protection character (18 x 10)
20	Belarusian product identification code (footwear, tires, consumer goods)
21	Protected physical data carrier with an identification means, 28x30 mm

### 4.2.3 Directory "Label statuses"

Class name: LabelStatus

Parameters:

**Table 4.2.3.1** LabelStatus class metadata structure

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

Values of the directory "Label statuses" are given in Table [4.2.3.2](#).

**Table 4.2.3.2** Label status list

code / Code	name / Name
0	Generated
10	Label ordering
15	Label issued
20	Manufactured
30	Validated
40	Signed in the Russian Federation
45	Imported
47	Applied
50	Item labeled
52	Transit item
55	In circulation in the Republic of Belarus
80	Sold in retail
90	Left the country

code / Code	name / Name
100	Return to retail
200	Invalid. Discarded

#### 4.2.4 Directory "Order statuses"

Class name: OrderStatus

Parameters:

**Table 4.2.4.1** OrderStatus class metadata structure

Key	Data type	Description
code	Integer	Code (identifier) of the order status
name	String	Order status name

Values of the directory "Order statuses" are given in Table [4.2.4.2](#).

**Table 4.2.4.2** Order status list

code / Code	name / Name
0	Created
10	Accepted from a business entity and entered on queue
15	Sent to the Russian Federation for signature (pending)
20	The Russian Federation has issued codes
30	Completed
40	Canceled

#### 4.2.5 Directory "Report statuses"

Class name: ReportStatus

Parameters:

**Table 4.2.5.1** ReportStatus class metadata structure

Key	Data type	Description
code	Integer	Code (identifier) of the report status
name or message	String	Report status name
reason	String	Error description

Values of the directory "Report statuses" are given in Table [4.2.5.2](#).

**Table 4.2.5.2** Report status list

code / Code	name / Name
0	Created
10	In process
20	Confirmed
25	Completed partially
30	Completed
40	Error

#### 4.2.6 Directory "Shipment statuses"

Class name: ShipStatus

Parameters:

**Table 4.2.6.1** ShipStatus class metadata structure

Key	Data type	Description
code	Integer	Code (identifier) of the shipment status
name	String	Shipment status name

Values of the directory "Shipment statuses" are given in Table [4.2.6.2](#).

**Table 4.2.6.2** Shipment status list

code / Code	name / Name
0	Created
5	In process
7	Rejected
10	En route
12	Shipped
15	Error
17	Pending acceptance
19	Capitalized partially
20	Accepted
30	Canceled

#### 4.2.7 Directory "Participant roles"

Class name: UserRole

Parameters:

**Table 4.2.7.1** UserRole class metadata structure

Key	Data type	Description
code	String	Role code (identifier)
description	String	Role name (description)

Some values of the directory "Participant roles" are given in Table [4.2.7.2](#).

**Table 4.2.7.2** List of participant roles (fragment)

code / Code	description / Description
exporter	Circulation participant
head	Head office
ukz	UCM circulation participant

#### 4.2.8 Directory "Report types"

Class name: ReportType

Parameters:

**Table 4.2.8.1** ReportType class metadata structure

Key	Data type	Description
id	Integer	Code (identifier) of the report type
name	String	Report type name

Values of the directory "Report types" are given in Table [4.2.8.2](#).

**Table 4.2.8.2** Report type list

id / Code	name / Name
1	Circulation
2	Aggregation
4	UCM usage
5	Preparation for circulation participant
6	Circulation. Residual stock
7	Circulation in the Republic of Belarus
8	Cancellation
10	Labeling of goods made from customer-owned raw materials
11	Discarding
13	Labeling parameter change
14	Manufacturing parameter report
15	Owner change
16	Deletion of codes from aggregate

#### **4.2.9 Directory "Import statuses"**

Class name: ImportStatus

Parameters:

**Table 4.2.9.1** ImportStatus class metadata structure

Key	Data type	Description
code	Integer	Code (identifier) of the supply status
name	String	Import status name

Values of the directory "Import statuses" are given in Table [4.2.9.2](#).

**Table 4.2.9.2** Import status list

code / Code	name / Name
0	Created
5	In process
10	Processed
20	Capitalization in process
30	Capitalized
40	Error

#### **4.2.10 Directory "Request statuses"**

Class name: RequestStatus

Parameters:

**Table 4.2.10.1** RequestStatus class metadata structure

Key	Data type	Description
code	Integer	Code (identifier) of the request status
name	String	Request status name

Values of the directory "Request statuses" are given in Table [4.2.10.2](#).

**Table 4.2.10.2** Request status list

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

### 4.3 Enumerations

#### 4.3.1 Enumeration "Countries"

List number: country

Some values of the enumeration "Countries" are given in Table [4.3.1](#).

**Table 4.3.1** Country list (fragment, EAEU countries)

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

#### 4.3.2 Enumeration "Currencies"

List number: currency

Some values of the enumeration "Currencies" are given in Table [4.3.2](#).

**Table 4.3.2** Currency list (fragment)

code / Code	name / Name
byn933	Belarusian ruble
eur978	Euro
kzt398	Kazakhstani tenge
rub643	Russian ruble
usd840	US dollar

#### 4.3.3 Enumeration "Document types"

List number: shippingDocs

Some values of the enumeration "Types" are given in Table [4.3.3](#).

**Table 4.3.3** Document type list (fragment)

code / Code	name / Name
akt	Certificate
aktnedost	Statement of shortage
aktreorg	Deed of conveyance
invalid_doc	Notification of invalidation
invoice	Invoice, bill
customs	Customs declaration
statdeklaracia	Statistical declaration
ttn	Waybill (CMR)
tnttn	Delivery slip
ettn	Electronic waybill (eCMR)
etn	Electronic delivery slip

#### **4.3.4 Enumeration "Discard reasons"**

List number: reasons

Some values of the enumeration "Discard reasons" are given in Table [4.3.4](#).

**Table 4.3.4** Discard reason list (fragment)

code / Code	name / Name
damage	Deterioration
destruction	Destruction
liquidation	Winding up (cessation of business)
lost	Loss
mns	Notification from the supervisory authority (Ministry of Taxation)
other	For manufacturing needs
spoilage	Manufacturing defect
theft	Theft
usemark	Use (labeling)

#### **4.3.5 Enumeration "Circulation reasons"**

List number: circulation\_reason

Some values of the enumeration "Circulation reasons" are given in Table [4.3.5](#).

**Table 4.3.5** List of circulation reasons

code / Code	name / Name
import	Import from EAEU countries
import_third_countries	Import from third countries

#### **4.3.6 Enumeration "Certificate types"**

List number: certificate\_types

Values of the enumeration "Certificate types" are given in Table [4.3.6](#).

**Table 4.3.6** List of types of conformity documents (types of authorization documents and certificates)

code / Code	name / Name
CONFORMITY_CERTIFICATE	Conformity certificate
CONFORMITY_DECLARATION	Conformity declaration
STATE_REGISTRATION_CERTIFICATE	State registration certificate
REGISTRATION_VET_CERTIFICATE	Registration certificate (for veterinary use)

#### 4.3.7 Enumeration "Owner change reasons"

List number: change\_of\_owner

Values of the enumeration "Owner change" are given in Table [4.3.7](#).

**Table 4.3.7** Owner change reason list

code / Code	name / Name
reorganization	Reorganization
change_owner_goods	Product owner change

## 5 ERROR CODES AND THEIR DESCRIPTIONS

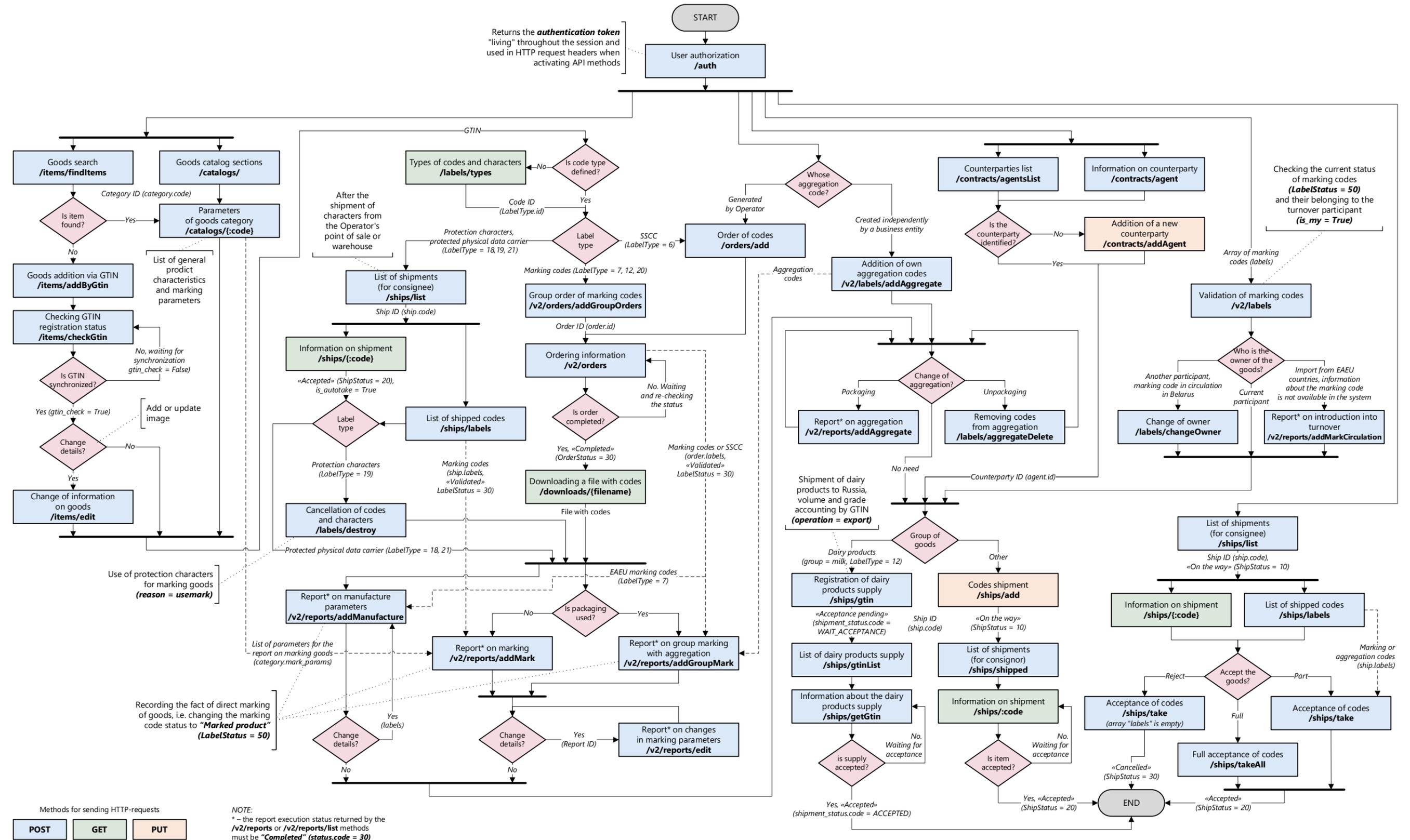
The HTTP response may return error codes with corresponding descriptions listed in Table [5](#).

**Table 5** Error codes and their descriptions

Error code	Description
<b>HTTP response code 400. Invalid data</b>	
400	Error in the input parameters of the request
<b>HTTP response code 401. Invalid authentication token</b>	
1202	Invalid username
1203	Invalid password
10004	Authentication token expired
13001	Invalid token
13002	Invalid token or the user is blocked
<b>HTTP response code 403. No access rights</b>	
403	File not found or you do not have access rights
13003	Access denied / Access prohibited
2102	You cannot create this order
<b>HTTP response code 404. Requested data not found</b>	
1250	Enumeration list not found
1401	No requests found
1451	Agent not found
1452	Unable to add agent
1453	Contract (agreement) not found
1454	Unable to update contract (agreement) data

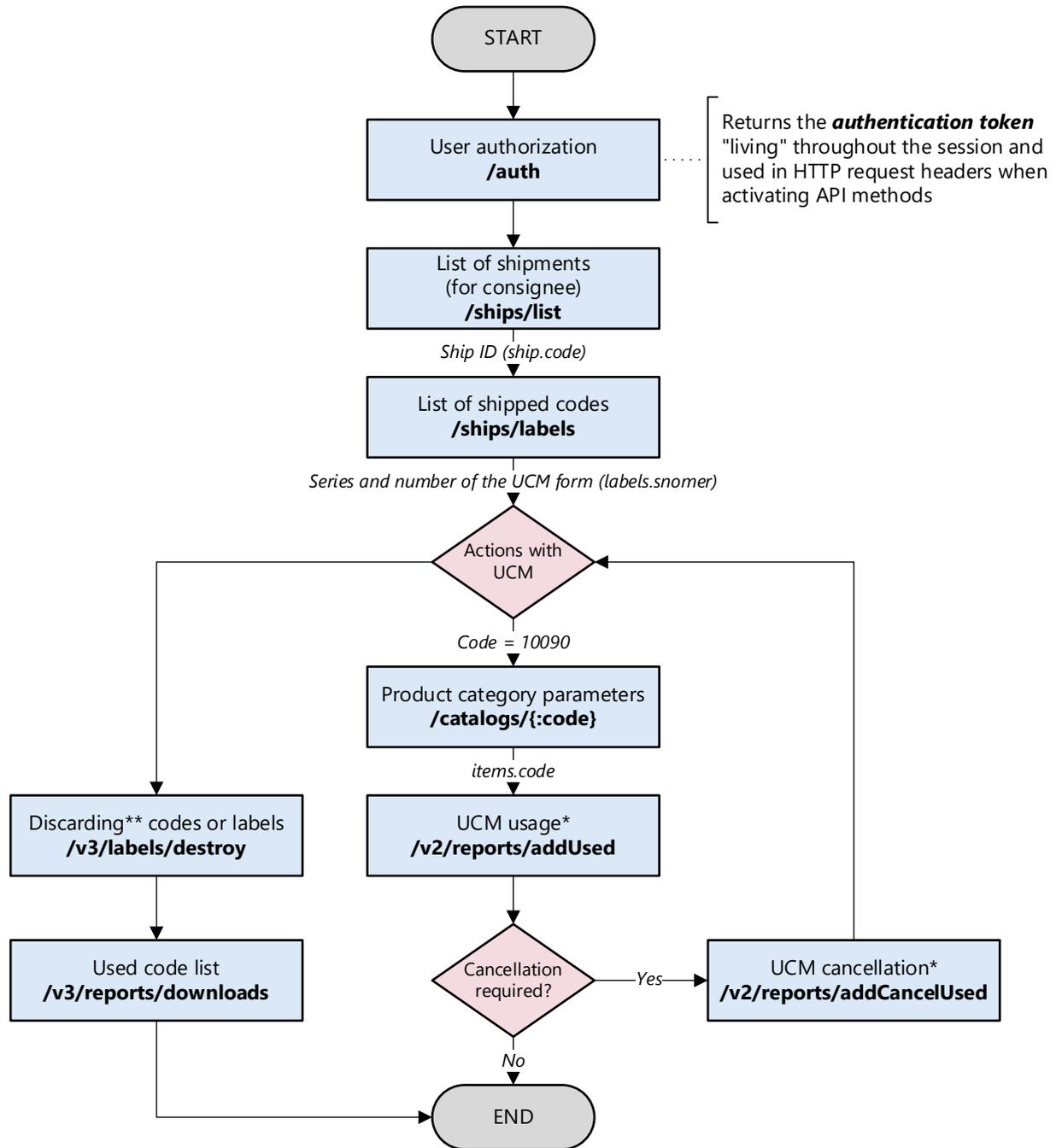
<b>Error code</b>	<b>Description</b>
1455	An agent with this data already exists in the system
1450	Unable to update code statuses
1488	Shipment not found
1501	Order not found
1507	Unable to save codes from order
1508	Product group error
1509	You must be the owner of the codes. Code status must be "Code Issued". GTIN must correspond to the product group
1660	Product catalog section not found
1701	Labeling code or aggregation code not found
1702	Error creating/updating code details
1706	The specified code is not an aggregation code
1708	Invalid code type
1711	Unable to add aggregation code
1766	Report not found
2001	Product not found
2004	Code type not found
2005	Error updating product details
2007	Invalid product parameters
2008	Unable to save product parameters
2010	Product GTIN not found
2011	EAEU CN of FEA code not specified
2012	Required characteristics of the product are not specified
10001	Supply document not found or already processed

### ANNEX A Sequence diagram of calling API methods when labeling goods with identification means



## ANNEX B

### Sequence diagram of calling API methods when labeling goods with unified control marks



**NOTES:**

\* – the report execution status returned by the **/v2/reports** or **/v2/reports/list** methods must be **"Completed"** (**status.code = 30**)

\*\* – the report execution status is checked by the **/v3/reports/list** method

Methods for sending HTTP-requests

**POST**

## ANNEX C

### Description of labeling business processes in individual product groups

#### C.1 General specifics of labeling of goods with identification means

For items for which the date of commencement of mandatory labeling with IM in the territory of the Republic of Belarus has not been reached or such labeling is not applied but has been implemented in other EAEU member states, the basic API methods are used in general, called in the following sequence:

##### *C.1.1 Representation A*

C.1.1.1 The method `/v2/orders/addGroupOrders` (see [3.3.4.1](#)) with the code type `label_type = 7` and specified value of the parameter "group" executes a group order of LCs for the whole required nomenclature of products in the required volume for each item. Upon successful execution of the LC order, the status "Item labeled" (`status = 50`) is assigned, except for non-alcoholic beverages and juices, for which the intermediate status "Applied" (`status = 47`) is assigned.

C.1.1.2 The method `/v2/orders/statusList` (see [3.3.5](#)) checks the current status of the created orders.

C.1.1.3 For orders with status "Complete" (`status.code = 30`), the `/v2/orders` method (see [3.3.2.1](#)) returns an array of generated LCs by order ID.

C.1.1.4 Execution of a labeling report is not required.

C.1.1.5 If necessary, before shipment of labeled goods by `/v2/reports/addChangeOwnerMade` method (see [3.5.15.1](#)), the current owner of LCs may be changed, provided that these LCs do not belong to a circulation participant (business entity) and the data on such LCs are available in SIS "Electronic Mark".

C.1.1.6 The method `/ships/add` (see [3.7.1.1](#)) sends information about shipment (transfer, transportation) of labeled goods by the shipper to the agent (business entity), including export.

C.1.1.7 Methods `/ships/take` (see [3.7.4.1](#)) or `/ships/takeAll` (see [3.7.12.1](#)) are used by the recipient once to accept all or a part of the labeled items or reject the shipment.

##### *C.1.2 Representation B*

C.1.2.1 The method `/v3/orders/addGroupOrders` (see [3.3.4.2](#)) with the code type `label_type = 7` and specified value of the parameter "group" executes a group order of LCs for the whole required nomenclature of products in the required volume for each item. Upon successful execution of the LC order, the status "Applied" (`status = 47`) is assigned.

C.1.2.2 The method `/v3/orders/list` (see [3.3.3.2](#)) checks the current status of the created orders.

C.1.2.3 For orders with status "Complete" (`status.code = 30`), the method `/v3/orders/downloads` (see [3.3.7](#)) returns an array of generated LCs by the file name retrieved from the order (`files.labels.filename`).

C.1.2.4 Execution of a labeling report is not required.

C.2.1.5 If necessary, before shipment of labeled goods by `/v3/reports/addChangeOwnerMade` method (see [3.5.15.2](#)), the current owner of codes may be changed, provided that these LCs or ACs do not belong to a circulation participant (business entity) and the data on such codes are available in SIS "Electronic Mark".

C.1.2.6 The method `/v3/ships/add` (see [3.7.1.2](#)) sends information about shipment (transfer, transportation) of labeled goods by the shipper to the agent (business entity), including export.

C.1.2.7 The method `/v3/ships/take` (see [3.7.4.2](#)) or `/v3/ships/takeAll` (see [3.7.12.2](#)) is used by the recipient once to accept all or a part of the labeled goods or reject the shipment.

## C.2 Specifics of labeling processes for dairy products, footwear, tires and consumer goods

For product groups "Dairy products" (group = milk), "Footwear" (group = shoes), "Tires" (group = tires) and "Consumer goods" (group = clothes), the following API methods are used in the specified sequence, taking into account the specifics described below:

C.2.1 The method `/v2/orders/addGroupOrders` (see [3.3.4.1](#)) with the code type `label_type = 12` (for dairy products) or `label_type = 20` (for footwear and tires) executes a group order of LCs for the whole required nomenclature of products in the required volume for each item. Upon successful execution of the LC order, the status "Validated" (`status = 30`) is assigned.

C.2.2 Using the label order method `/v2/orders/add` (see [3.3.1](#)) with the code type `label_type = 6`, the generation of ACs in SSCC format is carried out, or using the method `/v2/labels/addAggregate` (see [3.4.1](#)) custom ACs generated by the circulation participant (`label_type = 13`) are added.

C.2.3 The method `/v2/reports/addMark` (see [3.5.1](#)) is used for sending the labeling report, or the method `/v2/reports/addGroupMark` (see [3.5.2](#)) is used for sending the group labeling report (`report_type = 1`). In the latter case, simultaneously with the IM application report, the operation of aggregation of the first level packages is performed, i.e., the report on the application of IMs on products (use of LCs) and packages (use of ACs) and they are simultaneously linked. That is, this method combines methods `/v2/reports/addMark` (see [3.5.1](#)) and `/v2/reports/addAggregate` (see [3.5.3.1](#)). When processing reports, the code type (`label_type = 12, 18, 20` or `21`), the LC status "Validated" (`status = 30`) and the ownership by the circulation participant (i.e., the business entity that ordered and received the LC) are checked.

C.2.4 When sending a labeling report or a group labeling report (`report_type = 1`), an array of labeling parameters "params" must be specified, the list and identifiers (codes) of which and the types of their values can be retrieved from the attribute "mark\_params" returned in the response of the method `/catalogs/{:code}` (see [3.2.2](#)), i.e., from the request for information about the

corresponding section of the catalog (category) of items and the composition of its parameters.

C.2.5 List of labeling parameters (labeled item details) sent in the labeling report is given in Table [D.1](#) of Annex [D](#).

C.2.6 The method `/v2/reports` (see [3.5.5](#)) or `/v2/reports/list` (see [3.5.6.1](#)) can be used for checking the status by the labeling report UID (`report_type = 1`). Upon successful execution of the LC report, the status "Labeled" (`status = 50`) is assigned.

C.2.7 The method `/v2/reports/edit` (see [3.5.13](#)) allows to correct values of labeling parameters "params" of previously submitted labeling report/group labeling report (`report_type = 1`).

C.2.8 Method `/ships/gtin` (see [3.7.6](#)) without specifying the LC sends data on the supply (shipment) of dairy products to the Russian Federation in terms of GTIN and quantity (volume and sort accounting). The method `/ships/gtinList` (see [3.7.7](#)) returns a list of completed shipments.

### C.3 Specifics of labeling of goods with unified control marks

For items subject to UCM labeling (`group = ukz`), both general and specific API methods are used and called in the following sequence:

C.3.1 Methods `/ships/list` (see [3.7.5.1](#)) and `/ships/labels` (see [3.7.3.1](#)) are used to retrieve the list of series and numbers of UCM forms sold to the circulation participant.

C.3.2 The method `/catalogs/{:code}` (see [3.2.2](#)), where `code = 10090`, which corresponds to a request for information on the catalog section (category) of UCM labeled items, is used to return a list of relevant items and their codes in array "items".

C.3.3 The method `/v2/reports/addUsed` (see [3.5.4](#)) sends the report on the actual use of UCM (application to the item).

C.3.4 The method `/v2/reports/addCancelUsed` (see [3.5.11](#)) is used for full or partial cancellation of an UCM in case of a need to correct a previously submitted report.

C.3.5 The method `/v2/reports` (see [3.5.5](#)) or `/v2/reports/list` (see [3.5.6.1](#)) can be used for checking the state of a UCM usage report (`report_type = 4`) or a UCM cancellation report (`report_type = 8`) by the UID.

### C.4 Specifics of residual stock labeling

The following API methods are used for labeling of residual stock, which are called in a specific sequence and subject to the following specifics:

C.4.1 The method `/v2/items/inventory` (see [3.2.7](#)) with specifying the corresponding value of the parameter "group" is used for sending the results of the residual stock inventory.

C.4.2 The method `/v2/orders/addGroupOrders` (see [3.3.4.1](#)) with code type `label_type = 20` is used for ordering LCs in the required amount. In this case, either the GTIN for the residual stock (Table [C.4.1](#)) without their full

description is used if the residual stock of such goods is to be sold only in Republic of Belarus without export, or the GTIN for the corresponding nomenclature items with their full description is used. LC is assigned with the status "Validated" (`status = 30`).

**Table C.4.1** GTINs used for residual stock labeling

GTIN	Description
04814130000030	Residual stock. Footwear. This product is for marketing ONLY in the Republic of Belarus
04814130000047	Residual stock. Tires. This product is for marketing ONLY in the Republic of Belarus
04814130000054	Residual stock. Consumer goods. This product is for marketing ONLY in the Republic of Belarus

C.4.3 The method `/v2/orders/statusList` (see [3.3.5](#)) checks the current status of the created orders.

C.4.4 For orders with status "Complete" (`status.code = 30`), the `/v2/orders` method (see [3.3.2.1](#)) returns an array of generated LCs by order ID.

C.4.5 The method `/v2/orders/addMarkLeftovers` (see [3.5.10](#)) is used for sending the residual stock labeling report (`report_type = 6`). When sending reports, it is allowed to use LCs with label types `label_type = 7` or `20`. In this case, for LCs with `label_type = 20`, its ownership by the circulation participant (i.e., the business entity that ordered and received the LC) is additionally checked.

C.4.6 The method `/v2/reports` (see [3.5.5](#)) or `/v2/reports/list` (see [3.5.6.1](#)) can be used for checking the status by the residual stock labeling report UID (`report_type = 6`). Upon successful execution of the report, the LC is assigned with a new status "In circulation in the Republic of Belarus" (`status = 55`).

## C.5 Specifics of acceptance and circulation of goods purchased in cross-border trade

For goods labeled with IMs and purchased in other EAEU member states (in particular, in the Russian Federation) in course of cross-border trade, the operations of acceptance and circulation are carried out taking into account the following specifics:

C.5.1 The exporter (shipper) in the information goods labeling system of the EAEU member state from which the goods were imported, forms and sends to the importer (recipient) information on the supply of goods (cross-border shipment document), which is automatically received in SIS "Electronic Mark" (delivery type `type = ship`). The goods are assigned with the status "Acceptance pending".

C.5.2 Methods `/ships/importList` (see [3.7.8](#)) and `/ships/import` (see [3.7.10](#)) are used by the importer (recipient) for checking the availability and retrieving information on shipment (supply) of goods to their address.

C.5.3 The importer (recipient) enters information on actual acceptance of the goods purchased in a single or several transactions, dividing the supply into multiple parts (batches) in one of the following ways:

circulation by LC using method `/v2/reports/addMarkCirculation` (see [3.5.9](#));

Trust acceptance by identification codes using method `/ships/addImport` (see [3.7.9](#)). After that such goods can be put into circulation with LCs if necessary.

C.5.4 Regardless of the acceptance method:

the UID of the incoming document with supply type `type = export`, for which acceptance is performed, must be specified in the `"document_id"` attribute;

the acceptance operation is performed once for the entire document as a whole in one batch specifying all identification codes actually accepted from this supply; in this case the identification codes not listed in the trust acceptance or for which their corresponding LCs will not be specified in the circulation report are automatically rejected and returned to the supplier (i.e., the goods are not withdrawn from circulation in the territory of the Russian Federation, and the status "In circulation" is restored);

when attempting to send the information on acceptance of goods using the same UID of the supply document (`type = ship`) repeatedly, SIS "Electronic Mark" will generate an error on the missing supply document pending acceptance (error code `10001`).

C.5.5 After processing of data on cross-border shipment by the importer (recipient) and sending the result from SIS "Electronic Mark" to the information labeling system of the EAEU member state of the exporter (shipper), where the goods are assigned with the status "Exported to the EAEU" in case of successful acceptance or the status "In circulation" is restored in case of refusal of acceptance.

C.5.6 When using the `/v2/reports/addMarkCirculation` (see [3.5.9](#)), which specifies the reason "Import from EAEU countries" (`circulation_reason = import`), goods with IMs that arrived (imported) into the Republic of Belarus are put into circulation:

the `"params"` section of the circulation report must contain the following: the digital code of the country from which the goods were imported; the date and number of the document confirming the purchase of the goods; Taxpayer's Number or other equivalent identifier and the name of the supplier (seller organization);

since in shipments from the Russian Federation identification codes are sent without a crypto part ("crypto tail"), the list of accepted LCs must be specified in full format in the `"labels"` attribute of the circulation report;

it is allowed to put the goods into circulation in multiple steps, dividing the shipment into several parts (batches), provided that the trust acceptance of the entire cross-border shipment has been performed before or the goods have the status of "Exported to the EAEU" (in this case the `"document_id"` attribute is not filled).

C.5.7 The method `/v2/reports` (see [3.5.5](#)) or `/v2/reports/list` (see [3.5.6.1](#)) can be used for checking the status by the circulation report UID (`report_type = 7`). In case of successful completion of the operation of putting into circulation (capitalization) of goods, the LC is assigned with the status "In circulation in the Republic of Belarus" (`status = 55`).

## **C.6 Specifics of labeling of goods made from customer-owned raw materials**

For the labeling of goods made from customer-owned raw materials, operations are performed taking into account the following specifics:

C.6.1 LCs are generated by the customer (material supplier) in the information labeling system of the EAEU member state in which it is registered (e.g., in the Russian Federation) and sent to the contractor (processor, resident of the Republic of Belarus) under the contract concluded between them. In this case, the services of SIS "Electronic Mark" are not used.

C.6.2 The method `/v2/reports/addMarkTrans` (see [3.5.12](#)) is used by the processor for sending the labeling report regarding the goods made from customer-owned raw materials (`report_type = 10`).

C.6.3 The method `/v2/reports` (see [3.5.5](#)) or `/v2/reports/list` (see [3.5.6.1](#)) can be used for checking the status by the UID of the report on the labeling of goods made from customer-owned raw materials (`report_type = 10`). Upon successful execution of the LC report, the status "Transit item" (`status = 52`) is assigned.

C.6.4 Item with the current status "Transit item":  
is not subject to sale in the Republic of Belarus and is acceptable for circulation only in terms of transportation and storage;  
such items may be subsequently put into circulation as goods purchased in cross-border trade.

## **C.7 Specifics of labeling of goods that require the submission of information on manufacturing parameters**

### *C.7.1 Representation A*

For product groups "Packaged water" (`group = water`) and "Non-alcoholic beverages and juices" (`group = juice`), additional operations are performed prior to shipment of goods labeled with Russian IMs:

C.7.1.1 The method `/v2/reports/addManufacture` (see [3.5.14.1](#)) is used for sending the manufacturing parameter report (`report_type = 14`) with mandatory information on the manufacture (production) date in the parameter "manufacture\_date" and the expiration date in the parameter "expiration\_date".

C.7.1.2 The method `/v2/reports` (see [3.5.5](#)) or `/v2/reports/list` (see [3.5.6.1](#)) can be used for checking the status by the manufacturing parameter report UID (`report_type = 14`). Upon successful execution of the LC report, the status "Labeled" (`status = 50`) is assigned.

C.7.1.3 The manufacturing parameter report (`report_type = 14`) may be re-sent in order to correct previously submitted information.

### *C.7.2 Representation B*

For the product groups "Dietary supplements" (`group = dietary_supplements`), "Vegetable oils and fat-and-oil products"

(group = oil\_fat), "Animal feed" (group = animal\_feed), "Preserves" (group = preserves), "Veterinary medicines" (group = veterinary\_medicines), "Sturgeon roe and salmon roe" (group = caviar), "Bicycles and bicycle frames" (group = bike), "Antiseptics and disinfectants" (group = antiseptics), "Food products (Groceries)" (group = grocery), "Cosmetics and household cleaning products" (group = cosmetics), additional operations are performed prior to shipment of goods labeled with Russian IMs:

C.7.2.1 The method `/v3/reports/addManufacture` (see [3.5.14.2](#)) is used for sending the manufacturing parameter report (`report_type = 14`) with mandatory information on the manufacture (production) date in the parameter `"manufacture_date"`, the expiration date in the parameter `"expiration_date"` and/or other additional characteristics of the item, the list, identifiers and value groups of which can be retrieved from the attribute `"mark_params"` returned in the response of method `/catalogs/{:code}` (see [3.2.2](#)), i.e., from the request of information on the corresponding section of the item catalog (category) and its parameter set.

C.7.2.2 List of labeling parameters (labeled item details) sent in the manufacturing parameter report is given in Table [D.2](#) of Annex [D](#).

C.7.2.3 The method `/v3/reports/list` (see [3.5.6.2](#)) can be used for checking the status of the manufacturing parameter report by the UID (`report_type = 14`). Upon successful execution of the LC report, the status "Labeled" (`status = 50`) is assigned.

C.7.2.4 The manufacturing parameter report (`report_type = 14`) may be re-sent in order to correct previously submitted information.

## C.8 Specifics of business entity reorganization processes

Upon reorganization of a business entity, in order to transfer unused LCs, as well as UCMs, PPDCs and PCs (hereinafter "labels"), from the former owner (legal predecessor) to the new owner (legal successor), the latter shall perform the following operations:

### C.8.1 Representation A

C.8.1.1 The method `/v2/reports/addChangeOwnerMade` (see [3.5.15.1](#)) is used for transferring the report on owner change due to reorganization (`reason = reorganization`, `report_type = 15`) based on the information from the deed of conveyance or the division balance sheet indicating the previous owner, list of transferred LCs and/or labels, particulars of the document, based on which they are transferred.

C.8.1.2 The method `/contracts/agentsList` (see [3.6.3](#)) is used to search for the predecessor agent (previous owner), which was previously registered in SIS "Electronic Mark" as a circulation participant (`is_verified = True`) and may be a wound up business entity at the time of report submission.

C.8.1.3 The method `/v2/reports` (see [3.5.5](#)) or `/v2/reports/list` (see [3.5.6.1](#)) can be used for checking the status by the owner change report UID (in case

of reorganization, `report_type = 15`). Upon successful report execution, LCs and/or labels are assigned to the assignee (new owner).

### *C.8.2 Representation B*

C.8.2.1 The method `/v3/reports/addChangeOwnerMade` (see [3.5.15.2](#)) is used for transferring the report on owner change due to reorganization (`reason = reorganization`, `report_type = 15`) based on the information from the deed of conveyance or the division balance sheet indicating the previous owner, list of transferred LCs and/or labels (including in aggregates), particulars of the document, based on which they are transferred.

C.8.2.2 The method `/contracts/agentsList` (see [3.6.3](#)) is used to search for the predecessor agent (previous owner), which was previously registered in SIS "Electronic Mark" as a circulation participant (`is_verified = True`) and may be a wound up business entity at the time of report submission.

C.8.2.3 The method `/v3/reports/list` (see [3.5.6.2](#)) can be used for checking the status of the owner change report (in case of reorganization (`report_type = 15`) by the UID. Upon successful report execution, LCs and/or labels (including ACs) are assigned to the assignee (new owner).

**ANNEX D**  
**Parameters of labeling report in individual product groups**

**Table D.1** Item details sent in the labeling report

Labeling parameter <sup>1</sup>		Product group (Used <sup>2</sup> / Required <sup>3</sup> )			
code / Code	name / Name	Dairy products	Footwear	Tires	Consumer goods
<i>1. Mode "Manufacturing"</i>					
25	Manufacture (production) date	+ / Yes	+ / No	+ / No	+ / No
90	Batch number	+ / No	+ / No	+ / No	+ / No
91	Variable weight (value, unit) <sup>4</sup>	+ / Yes/No	–	–	–
92	Expiration date <sup>5</sup>	+ / Yes	–	–	–
93	Type of manufacturing order	+ / Yes	+ / Yes	+ / Yes	+ / Yes
94	Item name	+ / No	+ / No	+ / No	+ / No
95	Labeling method (polygraphic protection)	+ / Yes	+ / Yes	+ / Yes	+ / Yes
101	Labeling reason	–	+ / Yes, with value "1"	+ / Yes, with value "1"	+ / Yes, with value "1"
108	Dairy product labeling reason	+ / Yes, with value "1"	–	–	–
131	Veterinary accompanying document No.	+ / No	–	–	–
132	Veterinary accompanying document UI	+ / No	–	–	–
<i>2. Mode "Import from third countries"</i>					
25	Manufacture (production) date	+ / Yes	+ / No	+ / No	+ / No
90	Batch number	+ / No	+ / No	+ / No	+ / No
91	Variable weight (value, unit) <sup>4</sup>	+ / Yes/No	–	–	–
92	Expiration date <sup>5</sup>	+ / Yes	–	–	–
94	Item name	+ / No	+ / No	+ / No	+ / No
95	Labeling method (polygraphic protection)	+ / Yes	+ / Yes	+ / Yes	+ / Yes
96	Registration number of the item declaration <sup>6</sup>	+ / Yes	+ / Yes	+ / Yes	+ / Yes

Labeling parameter <sup>1</sup>		Product group (Used <sup>2</sup> / Required <sup>3</sup> )			
code / Code	name / Name	Dairy products	Footwear	Tires	Consumer goods
97	Date of release of goods in the item declaration	+ / Yes	+ / Yes	+ / Yes	+ / Yes
98	Contract number	+ / No	+ / No	+ / No	+ / No
99	Contract date	+ / No	+ / No	+ / No	+ / No
100	Exporting country	+ / Yes	+ / Yes	+ / Yes	+ / Yes
101	Labeling reason	–	+ / Yes, with value "2"	+ / Yes, with value "2"	+ / Yes, with value "2"
108	Dairy product labeling reason	+ / Yes, with value "2"	–	–	–
131	Veterinary accompanying document No.	+ / No	–	–	–
132	Veterinary accompanying document UI	+ / No	–	–	–
134	Serial number of goods in the item declaration	+ / No	+ / No	+ / No	+ / No
<b>3. Mode "Commission trade"</b>					
25	Manufacture (production) date	–	+ / No	+ / No	+ / No
90	Batch number	–	+ / No	+ / No	+ / No
94	Item name	–	+ / No	+ / No	+ / No
95	Labeling method (polygraphic protection)	–	+ / Yes	+ / Yes	+ / Yes
101	Labeling reason	–	+ / Yes, with value "3"	+ / Yes, with value "3"	+ / Yes, with value "3"
102	Basis: document date	–	+ / Yes	+ / Yes	+ / Yes
103	Basis: document number	–	+ / Yes	+ / Yes	+ / Yes
104	IM application reason	–	+ / Yes	+ / Yes	+ / Yes
<b>4. Mode "Relabeling"</b>					
25	Manufacture (production) date	+ / Yes	+ / No	+ / No	+ / No
90	Batch number	+ / No	+ / No	+ / No	+ / No
91	Variable weight (value, unit) <sup>4</sup>	+ / Yes/No	–	–	–
92	Expiration date <sup>5</sup>	+ / Yes	–	–	–

Labeling parameter <sup>1</sup>		Product group (Used <sup>2</sup> / Required <sup>3</sup> )			
code / Code	name / Name	Dairy products	Footwear	Tires	Consumer goods
94	Item name	+ / No	+ / No	+ / No	+ / No
95	Labeling method (polygraphic protection)	+ / Yes	+ / Yes	+ / Yes	+ / Yes
101	Labeling reason	–	+ / Yes, with value "4"	+ / Yes, with value "4"	+ / Yes, with value "4"
102	Basis: document date	+ / Yes	+ / Yes	+ / Yes	+ / Yes
103	Basis: document number	+ / Yes	+ / Yes	+ / Yes	+ / Yes
104	IM application reason	+ / Yes	+ / Yes	+ / Yes	+ / Yes
108	Dairy product labeling reason	+ / Yes, with value "4"	–	–	–
124	Relabeling reason	+ / No	+ / No	+ / No	+ / No
<b>5. Mode "Import from EAEU member states"<sup>7</sup></b>					
25	Manufacture (production) date	+ / Yes	–	–	–
90	Batch number	+ / No	–	–	–
91	Variable weight (value, unit) <sup>4</sup>	+ / Yes/No	–	–	–
92	Expiration date <sup>5</sup>	+ / Yes	–	–	–
94	Item name	+ / No	+ / No	+ / No	+ / No
95	Labeling method (polygraphic protection)	+ / Yes	+ / Yes	+ / Yes	+ / Yes
101	Labeling reason	–	+ / Yes, with value "5"	+ / Yes, with value "5"	+ / Yes, with value "5"
108	Dairy product labeling reason	+ / Yes, with value "5"	–	–	–
109	Date of the document confirming the purchase of the item	+ / Yes	+ / Yes	+ / Yes	+ / Yes
110	Number of the document confirming the purchase of the goods.	+ / Yes	+ / Yes	+ / Yes	+ / Yes
111	Tax number	+ / Yes	+ / Yes	+ / Yes	+ / Yes
112	Seller organization name	+ / Yes	+ / Yes	+ / Yes	+ / Yes
126	Exporting country (EAEU)	+ / Yes	+ / Yes	+ / Yes	+ / Yes

Labeling parameter <sup>1</sup>		Product group (Used <sup>2</sup> / Required <sup>3</sup> )			
code / Code	name / Name	Dairy products	Footwear	Tires	Consumer goods
131	Veterinary accompanying document No.	+ / No	–	–	–
132	Veterinary accompanying document UI	+ / No	–	–	–

**Notes:**

1. Parameter identifiers (codes) are different in the commercial and test sections. The table shows the parameter codes used in the commercial section.

2. Use of the labeling parameter when sending a labeling report for the corresponding product group and depending on the product use mode:

"+": used;

"-": not applied.

3. The necessity to fill the labeling parameter with the correct value when sending the labeling report:

Yes: required, subject to control;

No: no control.

4. Mandatory for goods with the variable weight flag.

5. The expiration date should not be earlier than the date of manufacture (production) of goods.

6. The registration number of the goods declaration should correspond to the established format:

[X...X/DDMMYY/NNNNNNN], where:

X...X: code of the customs authority which registered the item declaration in accordance with the classifiers of customs authorities applicable in the EAEU member states (in the Republic of Armenia: 2-digit code of the customs authority; in the Republic of Belarus, the Republic of Kazakhstan and the Kyrgyz Republic: 5-digit code; in the Russian Federation: 8-digit code);

DDMMYYYY: date of issue, signing, approval or registration of the document (day, month, last two digits of the year);

NNNNNNN: serial number of the customs document as per the registration log.

7. Applies in the case of import of items without IMs from an EAEU member state that has not implemented mandatory labeling of such goods.

**Table D.2** Item details sent in the manufacturing parameter report

Labeling parameter		Data type	Product group (Use <sup>1</sup> / Required <sup>2</sup> )									
slug / Key	name / Name		Dietary supplements	Vegetable oils and fat-and-oil products	Animal feed	Preserves	Veterinary medicines	Sturgeon roe and salmon roe	Bicycles and bicycle frames	Antiseptics and disinfectants	Food products (Groceries)	Cosmetics and household cleaning products
manufacture_date	Manufacture (production) date of goods	Date	+ / Yes	+ / Yes	+ / Yes	+ / Yes	+ / Yes	+ / Yes	+ / Yes	+ / Yes	+ / Yes	+ / Yes
expiration_date <sup>3</sup>	Expiration date	Date	+ / Yes	+ / Yes	+ / Yes	+ / Yes	+ / Yes	+ / Yes	-	+ / Yes	+ / Yes	+ / Yes
product_weight <sup>4</sup>	Actual item weight, g	Integer	-	-	+ / Yes/No	-	-	+ / Yes/No	-	-	-	-
nomer_serii	Series number	String	-	-	-	-	+ / Yes	-	-	-	-	-
alcohol_percentage	Actual content of ethyl alcohol per unit of product, %	String	-	-	-	-	-	-	-	-	-	+ / Yes <sup>5</sup>

**Notes:**

- Function of the labeling parameter when transmitting a manufacturing parameter report for the corresponding product group:  
"+": used;  
"-": not applied.
- The necessity to fill the labeling parameter with the correct value when sending the manufacturing parameter report:  
Yes: required, subject to control;  
No: no control.
- The expiration date should not be earlier than the date of manufacture (production) of goods.
- Mandatory for goods with the variable weight flag.
- The passed value is a number with one decimal places (digits) represented as a string without spaces in the format [0-9].[0-9]{2}. If absent, "0.0" is specified.

## ANNEX E

## Labeling code templates for individual product groups

Table E.1 List of labeling code templates established for individual product groups

id / Code	LC template structure	group / Product group	label_type / Label type	LC example
11	Standard LC: 01 + GTIN (14 digits) + 21 + serial number (13 characters) + 91 + verification key (4 characters) + 92 + verification code (44 characters)	bike ("Bicycles and bicycle frames")	7	0104814653133635215L;0kdrfsNFwo \u001d91FFD0\u001d92dGVzdEvN35Y AKoCjUzU2gdD36cU19yHMbVfdk/JuhJ w=
23	Short LC: 01 + GTIN (14 digits) + 21 + serial number (13 characters) + 93 + verification code (4 characters)	dietary_supplements ("Dietary supplements")	7	01006252730508822150g&aiuuY1)I5 \u001d93dGVz
25	Short LC: 01 + GTIN (14 digits) + 21 + serial number (13 characters) + 93 + verification code (4 characters)	antiseptics ("Antiseptics and disinfectants")	7	0104810874000770215PJF<ijtb(her \u001d93dGVz
26	Standard LC: 01 + GTIN (14 digits) + 21 + serial number (6 characters) + 91 + verification key (4 characters) + 92 + verification code (44 characters) <b>Note:</b> Primary (recommended) default template	animal_feed ("Animal feed")	7	0109143036545037215k9d8i\u001d9 1FFD0\u001d92dGVzdP46KSIh8+zqIR dwduX/vBafJsnihJ4vcvFcct8=
27	Standard LC: 01 + GTIN (14 digits) + 21 + serial number (6 characters) + 91 + verification key (4 characters) + 92 + verification code (44 characters) <b>Note:</b> Primary (recommended) default template	caviar ("Sturgeon roe and salmon roe")	7	0104815137001624215HTB0Y\u001d9 1FFD0\u001d92dGVzdA4kGOHIscdD7v OZ7XXjEKXaITct4JI05omCaf4=
30	Standard LC: 01 + GTIN (14 digits) +	dietary_supplements ("Dietary supplements")	7	0104812845008808215CjBmJL , pugAq \u001d91FFD0\u001d92dGVzdBSSkDJ

id / Code	LC template structure	group / Product group	label_type / Label type	LC example
	21 + serial number (13 characters) + 91 + verification key (4 characters) + 92 + verification code (44 characters) <b>Note:</b> Primary (recommended) default template			P1g/BAAbCv3wv/OaXx8Fb2u5JT9pZgKc=
31	Standard LC: 01 + GTIN (14 digits) + 21 + serial number (13 characters) + 91 + verification key (4 characters) + 92 + verification code (44 characters) <b>Note:</b> Primary (recommended) default template	antiseptics ("Antiseptics and disinfectants")	7	0104810874000770215<aSG5+++aaHAE\u001d91FFD0\u001d92dGVzdJJ15y66cZQ5bnSKdKmpSmxdwQ0cyMve/BeoG0c=
38	Short LC: 01 + GTIN (14 digits) + 21 + serial number (6 characters) + 93 + verification code (4 characters)	caviar ("Sturgeon roe and salmon roe")	7	0104815137001624215o8Cj!\u001d93dGVz
40	Short LC: 01 + GTIN (14 digits) + 21 + serial number (13 characters) + 93 + verification code (4 characters)	oil_fat ("Vegetable oils and fat-and-oil products")	7	0109002504112211215_odZ&Stvmlkp\u001d93dGVz
41	Short LC: 01 + GTIN (14 digits) + 21 + serial number (6 characters) + 93 + verification code (4 characters)	animal_feed ("Animal feed")	7	0106433036545001215t8dSH\u001d93dGVz
42	Standard LC: 01 + GTIN (14 digits) + 21 + serial number (13 characters) + 91 + verification key (4 characters) + 92 + verification code (44 characters) <b>Note:</b> Primary (recommended) default template	grocery ("Food products (Groceries)")	7	0104810005001256215T:e4Ehw:9hP_\u001d91FFD0\u001d92dGVzdLXNs68vuj2guqGJdS6VbN4uVbdQKn2v5dWae+k=
43	Short LC: 01 + GTIN (14 digits) +	grocery ("Food products	7	0104810005001256215yVPmInF3gkYV\u001d93dGVz

id / Code	LC template structure	group / Product group	label_type / Label type	LC example
	21 + serial number (13 characters) + 93 + verification code (4 characters)	(Groceries)"		
46	Standard LC: 01 + GTIN (14 digits) + 21 + serial number (6 characters) + 91 + verification key (4 characters) + 92 + verification code (44 characters) <b>Note:</b> Primary (recommended) default template	preserves ("Preserves")	7	0104810005001232215!DXum\u001d91FFD0\u001d92dGVzdPArfcUhw/zoODsOLWNTj650owM7E1rtOhPQRkE=
47	Short LC: 01 + GTIN (14 digits) + 21 + serial number (6 characters) + 93 + verification code (4 characters)	cosmetics ("Cosmetics and household cleaning products")	7	0104810005001232215XMVnQ\u001d93dGVz
48	Standard LC: 01 + GTIN (14 digits) + 21 + serial number (6 characters) + 91 + verification key (4 characters) + 92 + verification code (44 characters) <b>Note:</b> Primary (recommended) default template	cosmetics ("Cosmetics and household cleaning products")	7	0108855463017636215VpmWh\u001d91FFD0\u001d92dGVzdPgOu19XVnY2A1FSQ6TwBGGrmYZ/qJmoIrbP3ZU=
49	Short LC: 01 + GTIN (14 digits) + 21 + serial number (6 characters) + 93 + verification code (4 characters)	preserves ("Preserves")	7	0108855463017636215rHowY\u001d93dGVz
50	Standard LC: 01 + GTIN (14 digits) + 21 + serial number (13 characters) + 91 + verification key (4 characters) + 92 + verification code (44 characters)	veterinary_medicines ("Veterinary medicines")	7	0104028911553343215;hwq0.g6gnfk\u001d91FFD0\u001d92dGVzdIov8amsjmyNwEUYvqov45jAQW+lf6ZaEw1NdJ0=
51	Standard LC: 01 + GTIN (14 digits) + 21 + serial number (13 characters) + 91 + verification key (4 characters) +	oil_fat ("Vegetable oils and fat-and-oil products")	7	0105201409805158215'(e_lyDIYTD2\u001d91FFD0\u001d92dGVzdIPluA9yzkgDdoibb+5kavZiEK2PGQJp4BKSVXg=

<b>id / Code</b>	<b>LC template structure</b>	<b>group / Product group</b>	<b>label_type / Label type</b>	<b>LC example</b>
	92 + verification code (44 characters) <i>Note:</i> Primary (recommended) default template			

## ANNEX F Authorization documents

**Table F.1** Types of documents confirming compliance of goods with quality and/or safety requirements or registration of goods in accordance with the established procedure

group / Product group	certificate_type / Certificate type or document category (Used <sup>1</sup> / Required <sup>2</sup> )			
	CONFORMITY_CERTIFICATE ("Conformity certificate")	CONFORMITY_DECLARATION ("Conformity declaration")	STATE_REGISTRATION_CERTIFICATE ("State registration certificate")	REGISTRATION_VET_CERTIFICATE ("Registration certificate (for veterinary use)")
animal_feed ("Animal feed")	–	+ / No <sup>3</sup>	–	–
antiseptics ("Antiseptics and disinfectants")	+ / No	+ / No	+ / No	–
bike ("Bicycles and bicycle frames")	+ / No <sup>3</sup>	+ / No <sup>3</sup>	–	–
caviar ("Sturgeon roe and salmon roe")	+ / No	+ / No	+ / No	–
clothes ("Consumer goods")	+ / No	+ / No	+ / No	–
cosmetics ("Cosmetics and household cleaning products")	–	+ / No	+ / No	–
dietary_supplements ("Dietary supplements")	–	–	+ / No	–
grocery ("Food products (Groceries)")	–	+ / No	+ / No	–
juice ("Non-alcoholic beverages and juices")	+ / No	+ / No	–	–

group / Product group	certificate_type / Certificate type or document category (Used <sup>1</sup> / Required <sup>2</sup> )			
	CONFORMITY_CERTIFICATE ("Conformity certificate")	CONFORMITY_DECLARATION ("Conformity declaration")	STATE_REGISTRATION_CERTIFICATE ("State registration certificate")	REGISTRATION_VET_CERTIFICATE ("Registration certificate (for veterinary use)")
milk ("Dairy products")	+ / No	+ / No	+ / No	–
oil_fat ("Vegetable oils and fat-and-oil products")	+ / No <sup>3</sup>	+ / No <sup>3</sup>	+ / No <sup>3</sup>	–
perfum ("Perfume and eau de toilette")	+ / No	+ / No	+ / No	–
preserves ("Preserves")	+ / No <sup>3</sup>	+ / No <sup>3</sup>	+ / No <sup>3</sup>	–
photo ("Cameras and flash lamps")	+ / No	+ / No	–	–
shoes ("Footwear")	+ / No	+ / No	–	–
tires ("Tires")	+ / No	+ / No	–	–
veterinary_medicines ("Veterinary medicines")	–	–	–	+ / Yes
water ("Packaged water")	+ / No	+ / No	+ / No	–

**Notes:**

1. Use of the certificate type or document category when sending information on the shipment of labeled items in cross-border trade with the Russian Federation for the corresponding product group:

"+": used;

"-": not applied.

2. The requirement to fill in information on authorization documents when sending information on the shipment of labeled items in cross-border trade with the Russian Federation:

Yes: required, subject to control;

No: no control.

3. When shipping labeled goods to the Russian Federation, it is recommended that at least one of the allowed types of authorization documents is indicated.

**Table F.2** Number formats of documents confirming compliance of goods with quality and/or safety requirements or registration of goods in accordance with the established procedure

certificate_type / Type of certificate or document	Format of a document	
	issued in the Russian Federation	issued in other EAEU countries
CONFORMITY_ CERTIFICATE ("Conformity certificate")	<p>&lt;prefix&gt; RU C-XX.YYYY.Л.NNNNN/ГГ</p> <p>or</p> <p>TC RU C-XX.YYYY.Л.NNNNN</p> <p>where:</p> <p>&lt;prefix&gt;: type of the conformity certificate (Cyrillic letters), may be omitted; possible values: EAЭC, ПOCC, TC, BП;</p> <p>RU: two-letter alphabetic code of the Russian Federation (2 Latin letters);</p> <p>C: character indicating that the number belongs to a conformity certificate (a Cyrillic letter);</p> <p>XX: two-letter alphabetic code of the country of the manufacturer of the product for which the conformity certificate is issued (2 Latin letters);</p> <p>YYYY: four-letter code of the certification authority that issued the conformity certificate (4 digits and/or Cyrillic letters);</p> <p>Л: letter, code of type of certification object (a Cyrillic letter); possible values: A, batch or one-of-a-kind item, B, serial production;</p> <p>NNNNN: sequential number from 00001 to 99999;</p> <p>ГГ: last digits of the year of issue of the conformity certificate (2 digits from 00 to 99).</p>	<p>&lt;prefix&gt; N ZZ&lt;other characters&gt;</p> <p>where:</p> <p>&lt;prefix&gt;: type of the conformity certificate (Cyrillic letters), may be omitted; possible values: EAЭC, TC;</p> <p>N: number character, may be omitted; possible values: N, №;</p> <p>ZZ: two-letter alphabetic code of the EAEU member state where the certificate of conformity was issued (2 Latin letters); possible values: AM, BY, KG, KZ, RU;</p> <p>&lt;other character&gt;: 5 to 30 characters, including Latin and Cyrillic letters, digits, spaces and special characters (., -, _, /).</p> <p><b>Examples (illustrative):</b></p> <p>EAЭC BY/112 02.01. TP018 010.02 00259</p> <p>TC BY/112 03.11. 020.02 00031</p> <p>EAЭC AM-024/S.A-0247-2021</p> <p>EAЭC KZ 7500651.05.01.07531</p> <p>EAЭC KG417/016.CN.02.02302</p>

certificate_type / Type of certificate or document	Format of a document	
	issued in the Russian Federation	issued in other EAEU countries
	<p><b>Examples (illustrative):</b></p> <p>EAЭC RU C-CN.CБ21.A.00039/19  TC RU C-BY.MБ24.B.00534/18  POCC RU C-RU.MФ08.B.00237/18  RU C-RU.CT07.A.00007/19</p>	
CONFORMITY_ DECLARATION ("Conformity declaration")	<p>&lt;prefix&gt; RU Д-XX.YYYY.Л.NNNNN/ГГ  or  TC N RU Д-XX.YYYY.Л.NNNNN</p> <p>where:</p> <p>&lt;prefix&gt;: type of the conformity declaration (Cyrillic letters), may be omitted; possible values: EAЭC N, POCC, TC N, ВП;  RU: two-letter alphabetic code of the Russian Federation (2 Latin letters);  Д: character indicating that the number belongs to a conformity declaration (a Cyrillic letter);  XX: two-letter alphabetic code of the country of the manufacturer of the product for which the conformity declaration is registered (2 Latin letters);  YYYY: four-letter code of the authority that registered the conformity declaration (4 digits and/or Cyrillic letters);  Л: letter, code of type of certification object (a Cyrillic letter); possible values: A, batch or one-of-a-kind item, B, serial production;  NNNNN: sequential number from 00001 to 99999;  ГГ: last digits of the year of issue of the conformity declaration (2 digits from 00 to 99).</p>	<p>&lt;prefix&gt; N ZZ&lt;other characters&gt;</p> <p>where:</p> <p>&lt;prefix&gt;: type of the conformity declaration (Cyrillic letters), may be omitted; possible values: EAЭC, TC;  N: number character, may be omitted; possible values: N, №;  ZZ: two-letter alphabetic code of the EAEU member state where the conformity declaration was registered (2 Latin letters); possible values: AM, BY, KG, KZ, RU;  &lt;other character&gt;: 5 to 30 characters, including Latin and Cyrillic letters, digits, spaces and special characters (., -, _, /).</p> <p><b>Examples (illustrative):</b></p> <p>EAЭC № BY/112 11.02. TP021 013.01 03861  TC № BY/112 10.06. 002.03 00086  EAЭC AM-008/S.B-1397-2018  EAЭC KZ 4310317.13.12.00459  EAЭC KG417/533.Д.0006518</p>

certificate_type / Type of certificate or document	Format of a document	
	issued in the Russian Federation	issued in other EAEU countries
	<p><i>Examples (illustrative):</i></p> <p>EAЭС N RU Д-BY.PA01.B.35755/19  TC N RU Д-BY.PA01.B.00357/19  POCC RU Д-RU.PA01.B.22733/18  RU Д-RU.PA01.A.00021/18</p>	
STATE_ REGISTRATION_ CERTIFICATE ("State registration certificate")	<p>ZZ.11.YY.22.333.E.444444.MM.ГГ</p> <p>where:</p> <p>ZZ: two-letter alphabetic code of the EAEU member state where the state registration certificate was issued (2 Latin letters); possible values: AM, BY, KG, KZ, RU;</p> <p>11: two-digit numeric code of the region of the country or authority (2 digits from 01 to 99);</p> <p>YY: two-letter alphabetic or numeric code of the organization (2 digits from 01 to 99 or Cyrillic letters from AA to ЯЯ);</p> <p>22: two-letter numeric code of the workplace unique in this organization (2 digits);</p> <p>333 three-digit numeric code as per the unified product classifier (3 digits);</p> <p>E: letter; possible values: E, e, R, r (Latin letters) or E, e (Cyrillic letters);</p> <p>444444: six-digit numeric number of the state registration certificate issued in the current year in this organization (sequential number from 000001 to 999999);</p> <p>MM: sequential number of the month of issuance of the state registration certificate (2 digits from 01 to 12);</p> <p>ГГ: last digits of the year of issue of the state registration certificate (2 digits from 00 to 99).</p> <p><i>Examples (illustrative):</i></p> <p>BY.70.71.01.015.E.000634.10.24  RU.93.TC.12.005.E.000015.10.19</p>	

## REVISION HISTORY

Revision 1	Date 2	Change description 3	Author(s) 4
4.0	30.11.2024	<p>Document revision with amendments and additions.</p> <p>Added new version (v.3) API methods:</p> <ul style="list-style-type: none"> <li>- group order of LCs /v3/orders/addGroupOrders;</li> <li>- order details /v3/orders/list/{:id};</li> <li>- order list /v3/orders/list;</li> <li>- ordered code list /v3/orders/downloads;</li> <li>- label details /v3/labels;</li> <li>- manufacturing parameter report /v3/reports/addManufacture;</li> <li>- report list /v3/reports/list;</li> <li>- used code list /v3/reports/downloads;</li> <li>- code shipment /v3/ships/add;</li> <li>- shipment list /v3/ships/shipped and /v3/ships/list;</li> <li>- shipped code list /v3/ships/downloads;</li> <li>- code capitalization /v3/ships/take;</li> <li>- full acceptance of codes /v3/ships/take_all.</li> </ul> <p>New product groups added:</p> <ul style="list-style-type: none"> <li>- dietary_supplements ("Dietary supplements");</li> <li>- oil_fat ("Vegetable oils and fat-and-oil products")</li> </ul> <p>Added catalog sections:</p> <ul style="list-style-type: none"> <li>- 10097 ("Dietary supplements");</li> <li>- 10098 ("Vegetable oils and fat-and-oil products").</li> </ul> <p>Annex C updated according to the amendments and additions made.</p> <p>Added new Annexes E and F.</p>	V. N. Matokha
4.1	16.12.2024	<p>Document revision with amendments and additions.</p> <p>Removed description of the owner change method /labels/changeOwner.</p> <p>The functionality of method /v2/reports/addChangeOwnerMade has been extended allowing the use for changing the owner of a labeled item.</p> <p>An additional optional parameter "reason" has been added to the report list retrieval method /v2/reports/list to select owner change reports (report_type = 15) for the</p>	V. N. Matokha

Revision	Date	Change description	Author(s)
1	2	3	4
		corresponding owner change reason.	
4.2	19.12.2024	<p>Document revision with amendments and additions.</p> <p>The description of the <code>params</code> attribute of the manufacturing parameter report transmission method <code>/v3/reports/addManufacture</code> has been clarified.</p> <p>New product groups added:</p> <ul style="list-style-type: none"> <li>- <code>animal_feed</code> ("Animal feed");</li> <li>- <code>preserves</code> ("Preserves");</li> <li>- <code>veterinary_medicines</code> ("Veterinary medicines").</li> </ul> <p>Added catalog sections:</p> <ul style="list-style-type: none"> <li>- <code>10099</code> ("Preserves");</li> <li>- <code>10100</code> ("Animal feed");</li> <li>- <code>10101</code> ("Veterinary medicines").</li> </ul> <p>Annexes C, D and E updated according to the amendments and additions made.</p>	V. N. Matokha
4.3	31.12.2024	<p>Document revision with amendments and additions.</p> <p>For the method of adding custom aggregation codes <code>/v2/labels/addAggregate</code>, a restriction on the length of transmitted ACs (<code>label_type = 13</code>) of at least 21 characters has been added. The code ordering method <code>/v2/orders/add</code> with code type <code>label_type = 6</code> is recommended for generating ACs in SSCC format.</p> <p>Paragraph C.2.2 of Annex C was revised.</p>	V. N. Matokha
4.4	10.02.2025	<p>Document revision with amendments and additions.</p> <p>Added new version (v.3) API methods:</p> <ul style="list-style-type: none"> <li>- label ordering <code>/v3/orders/add</code>;</li> <li>- addition of custom aggregation codes <code>/v3/labels/addAggregate</code>;</li> <li>- aggregation report <code>/v3/reports/addAggregate</code>;</li> <li>- report on deletion of codes from the aggregate <code>/v3/reports/aggregateDelete</code>;</li> <li>- owner change report <code>/v3/reports/addChangeOwnerMade</code>;</li> <li>- label discarding <code>/v3/labels/destroy</code>;</li> <li>- information on the structure and composition of the aggregate <code>/v3/get_aggregate_inside</code>;</li> <li>- service delivery report list <code>/v3/common/getUsersBillingPdf</code></li> </ul>	V. N. Matokha

Revision	Date	Change description	Author(s)
1	2	3	4
		<p><b>ActsList.</b></p> <p>Added new report types:</p> <ul style="list-style-type: none"> <li>- type 11 ("Discarding");</li> <li>- type 16 ("Deletion of codes from the aggregate").</li> </ul> <p>Added description of metadata structure of a new object of <b>Aggregate</b> class.</p> <p>The description of the method for obtaining order information <code>/v3/orders/list/{:id}</code> is updated with examples of returned responses.</p> <p>In the method for obtaining a report list <code>/v3/reports/list</code>:</p> <ul style="list-style-type: none"> <li>- the request string is extended with additional optional selection parameters;</li> <li>- the composition of attributes returned in the response for different types of reports has been specified;</li> <li>- added additional examples of requests and responses generated for the corresponding report types.</li> </ul> <p>In the method for obtaining the used code list <code>/v3/reports/downloads</code>:</p> <ul style="list-style-type: none"> <li>- for attributes "report_id" and "report_uid", the list of methods that return values of report ID and unique report identifier, respectively, has been specified;</li> <li>- for attribute "file_type", the list of type values of returned information related to aggregates has been extended.</li> </ul> <p>The method for code shipment <code>/v3/ships/add</code> and the method for code capitalization <code>/v3/ships/take</code> were extended with the "aggregates" attribute for handling aggregates.</p> <p>The response returned by the method for obtaining the shipped code list <code>/ships/labels</code> is extended with the "parent" attribute which specifies the upper level AC containing the current LC or AC.</p> <p>The composition of attributes of class objects has been updated taking into account the differences in API methods in different versions:</p> <ul style="list-style-type: none"> <li>- Order (object "Order");</li> <li>- Ship (object "Shipment").</li> </ul> <p>The description of the methods for handling typography orders <code>/requests/addTypographyOrder</code> and</p>	

Revision	Date	Change description	Author(s)
1	2	3	4
		<p>/requests/listTypographyOrders, as well as the description of metadata structure of the TypographyOrder class object were removed.</p> <p>The name of the product group group = perfum and the corresponding catalog section 10087 has been changed to "Perfume and eau de toilette".</p> <p>In Annex C, paragraphs C.1 and C.8 have been updated according to the changes and additions made.</p> <p>In Annex F, Table F.2 has been added describing number formats of authorization documents.</p>	
4.5	28.02.2025	<p>Document revision with amendments and additions.</p> <p>New product groups added:</p> <ul style="list-style-type: none"> <li>- antiseptics ("Antiseptics and disinfectants");</li> <li>- bike ("Bicycles and bicycle frames");</li> <li>- caviar ("Sturgeon roe and salmon roe").</li> </ul> <p>Added catalog sections:</p> <ul style="list-style-type: none"> <li>- 10102 ("Sturgeon roe and salmon roe");</li> <li>- 10103 ("Bicycles and bicycle frames");</li> <li>- 10104 ("Antiseptics and disinfectants").</li> </ul> <p>Annexes C, D, E and F have been updated according to the changes and additions made.</p>	V. N. Matokha
4.6	01.04.2025	<p>Document revision with amendments and additions.</p> <p>The methods for sending a labeling report /v2/reports/addMark and a report on group labeling with aggregation /v2/reports/addGroupMark have a limitation on listing LCs of different types in the "labels" attribute.</p> <p>In the methods for obtaining a shipment list /v3/ships/shipped and /v3/ships/list the request string is extended with additional optional selection parameters.</p> <p>The composition of attributes of class objects Ship has been updated taking into account the differences in API methods in different versions.</p>	V. N. Matokha
4.7	04.04.2025	<p>Document revision with amendments and additions.</p> <p>New product groups added:</p> <ul style="list-style-type: none"> <li>- cosmetics ("Cosmetics and household cleaning products");</li> </ul>	V. N. Matokha

Revision	Date	Change description	Author(s)
1	2	3	4
		<p>- grocery ("Food products (Groceries)").</p> <p>Added catalog sections:</p> <ul style="list-style-type: none"> <li>- 10105 ("Food products (Groceries)");</li> <li>- 10106 ("Cosmetics and household cleaning products").</li> </ul> <p>Annexes C, D, E and F have been updated according to the changes and additions made.</p>	
4.8	30.04.2025	<p>Document revision with amendments and additions.</p> <p>An additional optional attributes "typography_id" and "typography_doc" has been added to the group order of LCs method /v3/orders/addGroupOrders to transfer the ordered LCs to the printing house that produces packaging materials.</p>	V. N. Matokha
4.9	28.05.2025	<p>Document revision with amendments and additions.</p> <p>The description of methods returning information about the code or label /labels/ and /v3/labels has been removed due to their decommissioning and termination of support from 01.07.2025.</p> <p>Annexes A and B have been updated according to the changes and additions made.</p>	V. N. Matokha
4.10	30.06.2025	<p>Document revision with amendments and additions.</p> <p>The functionality of the circulation report method /v2/reports/addMarkCirculation has been expanded in terms of sending information on goods marked with identification means imported from third countries.</p> <p>The methods for discarding codes or labels have been modified:</p> <ul style="list-style-type: none"> <li>- information about the use of PCs for labeling goods is sent using the /labels/destroy method;</li> <li>- the /v3/labels/destroy method is used to discarding codes or labels for other reasons.</li> </ul> <p><i>Note: The changes in the commercial section (production circuit) are effective from 14.07.2025.</i></p> <p>A new method that returns a list of accepted goods supplied from the EAEU /ships/importGtin has been added</p> <p>The temporary procedure for shipment of goods supplied from the territory of the Russian</p>	V. N. Matokha

<b>Revision</b>	<b>Date</b>	<b>Change description</b>	<b>Author(s)</b>
1	2	3	4
		Federation within the framework of cross-border trade has been excluded. Annexes B and C have been updated according to the changes and additions made.	