

API DESCRIPTION

Release 16.2

Table of content

1. General provisions	5
1.1 Interaction principles.....	5
1.2 Authentication	6
2. API authentication	6
2.1 API authentication.....	6
2.1.1 Method of random data output for signing with electronic signature	6
2.1.2 Method of obtaining an authentication security token	7
3 Marking codes and goods	9
3.1 Method for MC list output by a set filter with detailed information on MC	9
3.2 Method of detailed information receiving on particular MC	14
3.3 Method of output of a goods list by a set filter	16
3.4 Output of information on exact marking code	21
3.5 Method of receiving of brief information on MC\MC list (publicly available).....	25
4. Documents	27
4.1 Method of output of a list of documents previously uploaded to the TT IS	27
4.2 Method of content output of documents previously uploaded to the TT IS	32
4.3 Unified method of documents creation	35
4.4 Methods of documents creation	39
4.5 Method of draft document creation.....	41
4.6 Method of draft document receiving.....	43
4.7 Method of draft document update	44
4.8 Method of draft document deletion.....	45
4.9 Method of draft document searching and filtering.....	45
5. Goods guide GS1/National catalogue	46
5.1 Method of the GTIN list output for a turnover participant by INN	47
5.2 Method of information receiving on a goods item by GTIN	50
6 Guide of TNVED codes.....	53
6.1 Method of output of 10-character TNVED codes list.....	53
7 Appendix.....	54
8 Appendix. List of supported goods groups.....	55
9. Appendix. Document on release into turnover. Production in the Russian Federation.....	55

9.1 JSON format.....	55
10. Appendix. Document on release into turnover. Production outside EEU	57
10.1 JSON format.....	57
11 Appendix. Document on release into turnover. Contract production in the Russian Federation	58
11.1 JSON format.....	58
12 Appendix. Document on release into turnover. Receival from physical entities.	60
12.1 JSON format.....	60
13 Appendix. Document on release into turnover. Cross-border trade	61
13.1 JSON format.....	61
14 Appendix. Document on release into turnover. Marking of remaining amount.....	63
14.1 JSON format.....	63
15 Appendix. Document on goods shipment.....	64
15.1 JSON format.....	64
16 Appendix. Document on goods shipment cancellation	66
16.1 JSON format.....	66
17 Appendix. Document on goods acceptance.....	67
17.1 JSON format.....	67
18 Appendix. Document on goods withdrawal from turnover at sale to the final consumer	68
18.1 JSON format.....	68
19 Appendix. Document on MC drop out with/without IM	70
19.1 JSON format.....	70
20 Appendix. Document on goods remarking	71
20.1 JSON format.....	71
21 Appendix. Document on shipping package aggregation	72
21.1 JSON format.....	72
22 Appendix. Document on shipping package disaggregation.....	73
22.1 JSON format.....	73
23 Appendix. Document on shipping package re-aggregation.....	73
23.1 JSON format.....	73
24 Appendix. Document on remaining goods description	74
24.1 JSON format.....	74
25 Examples.....	75

25.1 Authentication by EQES/Documents signing	75
25.1.1 Example of a security token obtaining at login with attached signature/signing of documents with detached signature in 1C	75
25.1.2 Example of a security token obtaining at REACT JS login.	77
25.1.3 Example of a security token obtaining at login with attached signature/signing of documents with a detached signature on Java.	77

1. General provisions

1.1 Interaction principles

The interaction of the experiment participants and the TT IS operator is implemented electronically. The paper documentation exchange is not foreseen. The electronic interaction is performed using one of the following options:

1. Submission of information. Exchange of soft copies that are the base for amendments to the TT IS registers. Submission of information is defined as the obligation of participants.
2. Information request. The exchange when the information from the TT IS registers is provided to the participants in accordance with the established permits on information receiving; no amendments are to be incorporated into the registers. The request and the further submission of information from TT IS is initiated by a participant at own and sole discretion and applied, for instance, to verify marking codes statuses or participant's registration status.

The electronic interaction is performed by the exchange with soft copies in accordance with the established regulation. Enhanced and qualified electronic signature (EQES) is used for legally applicable electronic documents.

The electronic interaction is performed by means of the TT IS application programming interfaces (API) and information systems of participants.

The interaction is provided via the “client-server” model, where the TT IS is a server and the software of a participant is the client.

At interaction via API the following requirements are fulfilled:

1. Authentication and login of a participant (definition of jurisdictions for implementation of electronic interaction) is performed by means of EQES.
2. When submitting and requesting the information, the protection is performed via TLS/HTTPS

(GOST 28147-89) via encrypted data transmission channel to the Operator by CIPF certified in the Russian Federation (GOST R 34.10-2012).

The link for the TT IS industrial version: ismp.crpt.ru, the link for the TT IS demo version: demo.fashion.crpt.ru.

TT IS REST API responses can contain additional auxiliary json attributes that are not specified in the documentation. Unknown attributes shall be ignored.

Json of RFC 7159 standard is used for exchange with the TT IS (<https://tools.ietf.org/html/rfc7159>)

The maximum document size transmitted via API shall not exceed 70 MB.

All values of attributes in json and cvs formats that are valued in money (prices, taxes, etc.) shall be indicated in kopecks.

1.2 Authentication

NOTE:

All API methods of interaction with the TT IS require passing of the Authorization: Bearer <token> parameter in the header where token is an authentication token.

To receive an authentication security token of a turnover participant in the TT IS, API authentication and registration by means of EQES are implemented (see clause 2).

2. API authentication

2.1 API authentication

To receive an authentication token, it is required to request in the TT IS an array of random data (GET /api/v3/auth/cert/key), sign them by the EQES and send to the TT IS for verification (POST /api/v3/auth/cert/). If signature verification is successful, the TT IS returns an authentication token in jwt format (see jwt.io)

2.1.1 Method of random data output for signing with electronic signature

URL: /api/v3/auth/cert/key

Method: GET

Request example:

```
curl -v https://ismp.crpt.ru/api/v3/auth/cert/key
```

```
OTBet:
< HTTP/1.1 200
< Server: nginx/1.14.0
< Date: Fri, 25 Jan 2019 15:49:47 GMT
< Content-Type: application/json;charset=UTF-8
< Transfer-Encoding: chunked
< Connection: keep-alive
<
* Connection #0 to host ismp.crpt.ru left intact
```

```
{ "uuid": "b7b1abc9-f4ee-47db-8a20-f80ac83504e8", "data": "QNRPNPFGJZFUXCERQMTWLRMBRNRAAP" }
```

Response parameters:

Parameter	Type	Description
uuid	string	Unique identifier with generated random data
data	string	Random data string

Response example:

```
{
    "uuid": "<GUID>",
    "data": "<RANDOM STRING>"
}
```

http response status in case of success: 200 - OK

2.1.2 Method of obtaining an authentication security token

URL: /api/v3/auth/cert/key

Method: POST

Request header parameters:

- Authorization: Bearer <token>
- Content-Type: application/json

Request body parameters:

Parameter	Type	Necessity	Description
uuid	string	+	Unique identifier of signed random data (uuid from GET method /api/v3/auth/cert/key)
data	string	+	Random data signed by the EQES of the registered participant in base64 (attached electronic signature)

Body request:

```
{
    "uuid": "<GUID>",
    "data": "<Подписанные данные в base64 (ЭП присоединенная)>"
}
```

Response parameters:

Parameter	Type	Description
token	string	Authentication token in JWT format (https://jwt.io/)

Response:

```
{
    "token": "<JWT token>"
}
```

JWT token parameters:

Parameter	Type	Description
user_status	string	current status of the user in the TT IS
full_name	string	user full name
scope	array	TT IS auxiliary attribute
inn	string	INN of the participant
id	number	User registration number in the TT IS
pid	number	
position	string	Title of the participant
exp	number	Expiration time, Unix-time format
organization_status	string	User registration status in the TT IS
authorities	array	TT IS auxiliary attribute
jti	string	TT IS auxiliary attribute
client_id	string	TT IS auxiliary attribute

JWT token example:

```
{
    "user_status": <STRING>, пример: "ACTIVE",
    "full_name": <STRING>, пример: "Тестова Вера Владимировна",
    "scope": <МАССИВ СТРОК>, пример [ "trusted" ],
    "inn": <STRING>, пример: "6820022500",
    "id": <NUMBER>, пример: 50,
    "pid": <NUMBER>, пример: 12
    "position": "<STRING>", пример: Руководитель
    "exp": <NUMBER>, пример: 1548525191,
    "organization_status": <STRING>, пример: "REGISTERED"
    "authorities": <МАССИВ СТРОК>, пример: [ "0000000000",
"ROLE_ORG_TRADE_PARTICIPANT"],
    "jti": <STRING>,

```

```

"client_id": "crpt-service"
}

```

http response status in case of success: 200 - OK

Request example:

```
curl -X POST -v 'https://ismp.crpt.ru/api/v3/auth/cert/' -H 'content-type: application/json; charset=UTF-8' --data-binary '{"uuid": "50b47eeb-3dbc-46a8-beld-dac17449aaa4", "data": "<Подписанные данные в base64>"}
```

```

OTbet:
< HTTP/1.1 200
< server: nginx/1.14.1
< date: Sat, 26 Jan 2019 07:46:11 GMT
< content-type: application/json;charset=UTF-8
< vary: Origin
< vary: Access-Control-Request-Method
< vary: Access-Control-Request-Headers
< access-control-allow-origin: https://ismp.crpt.ru
< access-control-allow-credentials: true
<
* Connection #0 to host ismp.crpt.ru left intact
{"token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX3N0YXR1cyI6IkdVZlWRSIsImZlbGxibmFtZSI6IjE0LjQvZlVzcy0LaG0JLQtdGA0LaG0JLQu9Cw0LTQuNC80LjRgNC"}
-
0LLQvdCwIiwidXNlcl9uYW1lIjpudWxsLCJzY29wZSI6WyJ0cnVzdGVkIl0sImlubI6IjY4MjAwMjI1MDAiLCJpZCI6NTAsImV4cCI6MTU0ODUyNdc3MSwiYXV0aG9yaXRpZXMiOl0siSU5OXzY4MjAwMjI1MDAiLCJST0xFX09SR19UuKFERV9QVJUSUNJUEFOVCJdLCJqdGkiOiJmODI0ZWU3ZC1lMzh1LlTRkMDAtYmY2ZS1hMmE4NmJjYjMzOGQiLCJjbGllbnRfaWQiOiJjcnB0LXNlcnZpY2UifQ.mWffwihDEaYFzstRfIFd4Ni2wYN6fMY7VfCKJbbCjwU"}

```

3. Marking codes and goods

3.1 Method for MC list output by a set filter with detailed information on MC

The method returns a detailed list of marking codes that are currently available for the company (for the requesting company, a token contains INN) by a set filter.

URL: /api/v3/facade/identifytools/listV2

Method: GET

Request header parameters: Authorization: Bearer <token>

Request parameters:

Parameter	Type	Necessity	Description
-----------	------	-----------	-------------

Filtering			
cis	string	-	Marking code (both full and partial matches are admissible)
emissionDateFrom	string	-	Emission date “from” is specified in <i>yyyy-MM-dd'T'HH:mm:ss.SSS'Z</i> format Example: 2019-01-01T03:00:00.000Z
emissionDateTo	string	-	Emission date “to” is specified in <i>yyyy-MM-dd'T'HH:mm:ss.SSS'Z</i> format Example: 2019-01-10T03:00:00.000Z
gtin	string	-	Global Trade Item Number
sn	string	-	Serial number
tnVed10	string	-	10-character code of EEU TNVED
producerInn	string	-	INN of the producer
cisStatus	Array of string listings	-	Current status of MC. List of available values: • APPLIED - emitted.Issued • INTRODUCED - released into turnover • RETIRED - dropped out
emissionType	String listing	-	Emission type. List of available values: • LOCAL - production in the Russian Federation • FOREIGN - imported to the Russian Federation • REMAINS - marking of remaining amount • CROSSBORDER - imported from EEU countries
withdrawReason	string	-	Reason for MC withdrawal from turnover
packs	String listing	-	Display commodity packages only. List of available values: • ALL - Goods and packages • UNIT - Goods only • PACK - Packages only

cisAggregationState	String listing	-	MC status in an aggregate (packed/not packed/all). List of available values: • PACKED - Packed • NOT_PACKED - Not packed • ANY - Any
pg	string		Name of goods group (see the list of available goods groups in the “List of supported goods groups” appendix). Optional parameter.
Pagination/Sorting			
order	String listing	-	Sorting type (ascending/descending). List of available values: • ASC - ascending • DESC - descending
uit	String	-	MC value-"starting points" for records sorting.
orderedColumnValue	String	-	Column value-"starting points" for records sorting.
orderColumn	String listing	-	Column name used for sorting. List of available values: emd - emission date
pageDir	String listing	-	Selection of direction (forward/backward) for pagination. List of available values: • NEXT - forward • PREV - backward
limit	Integer	-	Maximum number of records that returns as a response.

Response parameters:

Parameter	Type	Description
results - list of information on MC		
cis	string	Marking code (MC)
gtin	string	Goods GTIN

tnVedEaes	number	10-character TNVED code
tnVedEaesGroup	number	4-character TNVED code
productName	string	Product name
emissionDate	string	MC emission date, yyyy-MM-dd'T'HH:mm:ss.SSS'Z' format
ownerInn	string	INN of the goods owner
ownerName	string	Name of the goods owner
status	string	MC status: • APPLIED - emitted.Issued • INTRODUCED - released into turnover • RETIRED - dropped out
emissionType	string	Production type: • LOCAL - production in the Russian Federation • FOREIGN - imported to the Russian Federation • REMAINS - marking of remaining amount • CROSSBORDER - imported from EEU countries
countChildren	number	Number of goods items in a packing.
packType	string	Packing type: UNIT - goods item, LEVEL1 - first level aggregate, LEVEL2 - second level aggregate, LEVEL3 - third level aggregate
parent	string	Parent aggregate code, if a goods item is in a packing
prevCis	string	Previous marking code value, if goods identification code was remarked

Response:

```
{
  "results": [
    {
      "cis": "<KM>",
      "gtin": "<GTIN>"
      "tnVedEaes": "<10-ти значный код ТН ВЭД>", пример "6401990000",
      "tnVedEaesGroup": "<4-х значный код ТН ВЭД>", пример "6405",
      "productName": "<Наименование товара>"
      "emissionDate": "<Дата эмиссии KM>", пример "2019-01-23T08:24:21.048Z",
      "ownerInn": "<ИНН собственника товара>"
      "ownerName": "<Наименование производителя>"
      "status": "<статус KM>", пример "INTRODUCED",
      "emissionType": "<тип производства>", пример "LOCAL"
    }
  ]
}
```

```

    }
  ],
  "total": 68
}

```

http response status in case of success: 200 - OK

Request example:

```

curl -X GET
'https://ismp.crpt.ru/api/v3/facade/identifytools/listV2?limit=10&order=DESC&
orderColumn=emd&uit=010469022978096621qSF7qjA3aRIqM2406402&orderedColumnValue
=2019-01-23T08:42:17.593Z&pageDir=NEXT' -H 'Authorization: Bearer <TOKEH>'

```

```

OTBET:
< HTTP/1.1 200
< Server: nginx/1.14.0
< Date: Sun, 27 Jan 2019 09:41:23 GMT
< Content-Type: application/json; charset=UTF-8
< Transfer-Encoding: chunked
< Connection: keep-alive
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1; mode=block
< Cache-Control: no-cache, no-store, max-age=0, must-revalidate
< Pragma: no-cache
< Expires: 0
< X-Frame-Options: DENY
<
* Connection #0 to host ismp.crpt.ru left intact
{
  "results": [
    {
      "cis": "00000000000000000000000000000000",
      "gtin": "04059155704529",
      "tnVedEaes": "6401990000",
      "tnVedEaesGroup": "6405",
      "productName": "Туфли женские, арт. 9-9-24302-21-019/6",
      "emissionDate": "2019-01-23T08:24:21.048Z",
      "ownerInn": "583713715202",
      "ownerName": "string",
      "participantName": "string",
      "participantInn": "0000000000",
      "status": "INTRODUCED",
      "lastDocId": "string",
      "emissionType": "LOCAL"
    }
  ]
  "total": 1
}

```

This request returns *10* (limit parameter) *next* (pageDir parameter) records starting from the emission date *2019-01-23T08:42:17.593Z* (orderedColumnValue parameter) and the marking code *010469022978096621qSF7qjA3aRIqM2406402* (uit parameter). The sorting will be performed by the emission date (orderColumn parameter) and (always) the marking code itself (regardless the transmitted parameters) by DESC (order parameter). The values of the emission

date and marking code shall be taken from the last string (for pageDir NEXT) and the first string (for pageDir PREV) to switch between pages.

3.2 Method of detailed information receiving on particular MC

The method returns detailed information on the requested MC (1 MC in 1 request) that is currently available for the company (the requesting company, a token contains INN).

URL: /api/v3/facade/identifytools/{cis}

Method: GET

URL parameters: cis - marking code.

Request header parameters: Authorization: Bearer <token>

Response parameters:

Parameter	Type	Description
results - list of information on MC		
cis	string	Marking code (MC)
gtin	string	Goods GTIN
tnVedEaes	number	10-character TNVED code
tnVedEaesGroup	number	4-character TNVED code
productName	string	Product name
emissionDate	number	MC emission date, yyyy-MM-dd'T'HH:mm:ss.SSS'Z' format
ownerInn	string	INN of a goods owner
ownerName	string	Name of a goods owner
status	string	MC status: • APPLIED - emitted.Issued • INTRODUCED - released into turnover • RETIRED - dropped out
emissionType	string	Production type: • LOCAL - production in the Russian Federation

		• FOREIGN - imported to the Russian Federation • REMAINS - marking of remaining amount • CROSSBORDER - imported from EEU countries
children	array of objects	List of MC in an aggregate, attribute content complies with the parent object
countChildren	number	Number of goods items in a packing.
packType	string	Packing type: UNIT - goods item, LEVEL1 - first level aggregate, LEVEL2 - second level aggregate, LEVEL3 - third level aggregate
parent	string	Parent aggregate code if a goods item is in a packing
prevCis	string	Previous marking code, if goods identification code was remarked

Response:

```
{
  "cis": "<KM>"
  "gtin": "<GTIN>"
  "tnVedEaes": "<10-ти значный код ТН ВЭД>", пример "6401990000",
  "tnVedEaesGroup": "<4-х значный код ТН ВЭД>", пример "6405",
  "productName": "<Наименование товара>"
  "emissionDate": "<Дата эмиссии КМ>", пример "2019-01-23T08:24:21.048Z",
  "ownerInn": "<ИНН собственника товара>"
  "ownerName": "<Наименование производителя>"
  "status": "<статус КМ>", пример "INTRODUCED",
  "emissionType": "<тип производства>", пример "LOCAL"
}
```

http response status in case of success: 200 - OK

Request example:

```
curl -X GET
'https://ismp.crpt.ru/api/v3/facade/identifytools/010405915570452921HXRNJW15a
sPkB2406405' -H 'Authorization: Bearer <TOKEH>'
```

```
Ответ:
< HTTP/1.1 200
< Server: nginx/1.14.0
< Date: Sun, 27 Jan 2019 09:41:23 GMT
< Content-Type: application/json; charset=UTF-8
< Transfer-Encoding: chunked
< Connection: keep-alive
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1; mode=block
< Cache-Control: no-cache, no-store, max-age=0, must-revalidate
< Pragma: no-cache
< Expires: 0
```

```

< X-Frame-Options: DENY
<
* Connection #0 to host ismp.crpt.ru left intact
{
  "cis": "00000000000000000000000000000000",
  "gtin": "04059155704529",
  "tnVedEaes": "6401990000",
  "tnVedEaesGroup": "6405",
  "productName": "string",
  "emissionDate": "2019-01-23T08:24:21.048Z",
  "ownerInn": "583713715202",
  "ownerName": "string",
  "participantName": "string",
  "participantInn": "0000000000",
  "status": "INTRODUCED",
  "lastDocId": "string",
  "emissionType": "LOCAL"
}

```

3.3 Method of output of a goods list by a set filter

URL: /api/v3/facade/marked_products/listV2

Method: GET

Request header parameters: Authorization: Bearer <token>

Request parameters:

Parameter	Type	Description
Filtering		
cis	string	Marking code (both full and partial matches are admissible)
emissionDateFrom	string	Emission date “from” is specified in yyyy-MM-dd'T'HH:mm:ss.SSS'Z format Example: 2019-01-01T03:00:00.000Z
emissionDateTo	string	Emission date “to” is specified in yyyy-MM-dd'T'HH:mm:ss.SSS'Z format Example: 2019-01-10T03:00:00.000Z
gtin	string	Goods GTIN
sn	string	Serial number
tnVed10	string	10-character EEU TNVED code
producerInn	string	INN of a producer

cisStatus	Array of string listings	Current status of a goods item. List of available values: • INTRODUCED - released into turnover • RETIRED - dropped out
emissionType	String listing	Emission type. List of available values: • LOCAL - production in the Russian Federation • FOREIGN - imported to the Russian Federation • REMAINS - marking of remaining amount • CROSSBORDER - imported from EEU countries
cisAggregationState	String listing	MC status in an aggregate (packed/not packed/all). List of available values: • PACKED - Packed • NOT_PACKED - Not packed • ANY - Any
statusExt	String listing	Additional status of a goods item. List of available values: • WAIT_SHIPMENT - waiting for acceptance confirmation • WAIT_TRANSFER_TO_OWNER - waiting for hand over to the owner • WAIT_REMARK - waiting for remarking
pg	String	Name of the goods group (see the list of available goods groups in “List of supported goods groups” appendix). Optional parameter.
Pagination/Sorting		
order	String listing	Sorting type (ascending/descending). List of available values: • ASC - ascending • DESC - descending
uit	string	MC value-"starting points" for records sorting.
orderedColumnValue	string	Column value-"starting points" for records sorting.

orderColumn	String listing	Column name used for sorting. List of available values: emd - emission date
pageDir	String listing	Selection of direction (forward/backward) for pagination. List of available values: • NEXT - forward • PREV - backward
limit	number	Maximum number of records that returns as a response.

Response parameters:

Parameter	Type	Description
results - list of information on MC		
cis	string	Marking code (MC)
gtin	string	Goods GTIN
tnVed	number	10-character TNVED code
productName	string	Product name
ownerName	string	Name of a goods owner company
ownerInn	string	INN of goods owner company
producerName	string	Name of a goods producer/importer
producerInn	string	INN of a goods producer/importer
status	string	Current status of a goods item. The list of available values is specified in the “Status and special goods condition” table.
statusEx	string	Special goods condition. The list of available values is specified in the “Status and special goods condition” table.
emissionDate		Production date <i>yyyy-MM-dd'T'HH:mm:ss.SSS'Z</i>
emissionType	string	Production type: • LOCAL - production in the Russian Federation • FOREIGN - imported to the Russian Federation • REMAINS - marking of remaining amount • CROSSBORDER - imported from EEU countries

lastDocId	string	Code of the last created document including the request
name	string	Goods name
brand	string	Trade brand
model	string	Goods producer (GS1 model)
prevCis	string	Previous marking code value (if a goods item was remarked)

The resulting JSON also can contain a set of fields special for particular goods item.

Status and special goods condition

Goods status	Special goods condition	status	statusEx
In turnover	-	INTRODUCED	NULL
In turnover	Waiting for acceptance confirmation	INTRODUCED	WAIT_SHIPMENT
In turnover	Waiting for hand over to the owner	INTRODUCED	WAIT_TRANSFER_TO_OWNER
In turnover	Waiting for remarking	INTRODUCED	WAIT_REMARK
Dropped out	Withdrawn by a receipt (it is a reason, not a special condition)	RETIRED	NULL

Response:

```
{
  "results": [
    {
      "cis": "00000000000000000000000000000000",
      "gtin": "04630034070029",
      "tnvd": "6401100000",
      "tnvedGroup": "6401",
      "productName": "string",
      "ownerName": "string",
      "ownerInn": "0000000000",

```

```

        "producerName": "string",
        "producerInn": "0000000000",
        "status": "INTRODUCED",
        "emissionDate": "2019-01-18T08:14:40.344Z",
        "lastDocId": "string",
        "name": "string",
        "brand": "отсутствует",
        "model": "CRPT--99999",
        "country": "РОССИЙСКАЯ ФЕДЕРАЦИЯ",
        "productTypeDesc": "ТАПОЧКИ",
        "color": "белый",
        "materialDown": "Текстиль",
        "productSize": "52",
        "materialUpper": "Текстиль",
        "materialLining": "Текстиль",
        "packageType": "BOX",
        "productType": "310000022"
    }
],
"total": 1
}

```

http response status in case of success: 200 - OK

Request example:

```

curl -X GET
'https://ismp.crpt.ru/api/v3/facade/identifytools/010405915570452921HXRNJW15a
sPkB2406405' -H 'Authorization: Bearer <TOKEH>'

```

```

Ответ:
< HTTP/1.1 200
< Server: nginx/1.14.0
< Date: Sun, 27 Jan 2019 09:41:23 GMT
< Content-Type: application/json; charset=UTF-8
< Transfer-Encoding: chunked
< Connection: keep-alive
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1; mode=block
< Cache-Control: no-cache, no-store, max-age=0, must-revalidate
< Pragma: no-cache
< Expires: 0
< X-Frame-Options: DENY
<
* Connection #0 to host ismp.crpt.ru left intact
{
  "results": [
    {
      "cis": "010463003407002921wskg1E44R1qym2406401",
      "gtin": "04630034070029",
      "tnvd": "6401100000",
      "tnvedGroup": "6401",
      "productName": "string",
      "ownerName": "string",
      "ownerInn": "0000000000",
      "producerName": "string",

```

```

    "producerInn": "0000000000",
    "status": "INTRODUCED",
    "emissionDate": "2019-01-18T08:14:40.344Z",
    "lastDocId": "string",
    "name": "string",
    "brand": "отсутствует",
    "model": "CRPT--99999",
    "country": "РОССИЙСКАЯ ФЕДЕРАЦИЯ",
    "productTypeDesc": "ТАПОЧКИ",
    "color": "белый",
    "materialDown": "Текстиль",
    "productSize": "52",
    "materialUpper": "Текстиль",
    "materialLining": "Текстиль",
    "packageType": "BOX",
    "productType": "310000022"
  }
],
"total": 1
}

```

This request returns *10* (limit parameter) *next* (pageDir parameter) records starting from the emission date *2019-01-23T08:42:17.593Z* (orderedColumnValue parameter) and the marking code *010463003407002921wskg1E44R1qym2406401* (uit parameter). The sorting will be performed by the emission date (orderColumn parameter) and (always) the marking code itself (regardless the transmitted parameters) by DESC (order parameter). The values of the emission date and marking code shall be taken from the last string (for pageDir NEXT) and the first string (for pageDir PREV) to switch between pages.

3.4 Output of information on exact marking code

URL: /api/v3/facade/marked_products/{uit}

Method: GET

URL parameters: uit - unique goods identifier (MC)

Request header parameters: Authorization: Bearer <token>

Response parameters:

Parameter	Type	Description
results - list of information on MC		
cis	string	Marking code (MC)
gtin	string	Goods GTIN
tnVed	number	4-character TNVED code

productName	string	Product name
ownerName	string	Name of a goods owner company
ownerInn	string	INN of a goods owner company
producerName	string	Name of a goods producer/importer
producerInn	string	INN of a goods producer/importer
status	string	Current status of a goods item. List of available values: • INTRODUCED - released into turnover • RETIRED - dropped out
emissionDate	string	Production date <i>yyyy-MM-dd'T'HH:mm:ss.SSS'Z</i>
emissionType	string	Production type: • LOCAL - production in the Russian Federation • FOREIGN - imported to the Russian Federation • REMAINS - marking of remaining amount • CROSSBORDER - imported from EEU countries
statusExt	string	Additional status of a goods item. List of available values: • WAIT_SHIPMENT - waiting for acceptance confirmation • WAIT_TRANSFER_TO_OWNER - waiting for hand over to the owner • WAIT_REMARK - waiting for remarking
withdrawReason	string	Withdrawal reason. The possible values are specified in the “Withdrawal reasons” table.
name	string	Goods name
brand	string	Trade brand
model	string	Goods producer (GS1 model)
certDoc		
type	string	Certificate type
number	string	Certificate number
date	string	Certificate date

country	string	Country of origin
productTypeDesc	string	Goods type
color	string	Colour
materialDown	string	Bottom material
productSize	string	Dimension
materialUpper	string	Top material
materialLining	string	Lining material
packageType	string	Packing type:
productType	string	Goods type

Withdrawal reason

Code	Value
RETAIL	Retail sale
EEC_EXPORT	Export to EEU countries
BEYOND_EEC_EXPORT	Export outside EEU countries
RETURN	Return to a physical entity
DAMAGE_LOSS	Loss or damage
DESTRUCTION	Destruction
CONFISCATION	Confiscation
LIQUIDATION	Enterprise liquidation
DONATION	Donation
STATE_ENTERPRISE	Purchase by a state enterprise
NO_RETAIL_USE	Usage for own needs by a buyer
ENTERPRISE_USE	Usage for own needs by an enterprise

Response:

```
{
  "cis": "<Уникальный идентификатор товара>"
  "gtin": "<gtin>"
  "tnvd": "<10-и значный код ТН ВЭД>", пример "6401100000",
  "tnvedGroup": <4-х значный код ТН ВЭД> "6401",
  "productName": "<Наименование товара>",
```

```

"ownerName": "<Наименование собственника товара>",
"ownerInn": "<ИНН собственника>"
"producerName": "<Наименование производителя/импортера товара>",
"producerInn": "<ИНН наименование производителя/импортера товара>"
"status": "<статус>", пример "INTRODUCED",
"emissionDate": "<дата эмиссии КМ>", пример "2019-01-18T08:14:40.344Z",
"producedDate": "<дата ввода товара в оборот>", пример "2019-01-
18T08:19:33.575Z",
"type": "<Тип КМ>", пример "PRINTED",
"prodOrderType": "<способ генерации серийного номера>", пример "OPERATOR",
"name": "Белые тапочки, арт. CRPT--99999, размер 52",
"brand": "отсутствует",
"model": "CRPT--99999",
"certDoc": {
  "type": "CONFORMITY_CERT",
  "number": "122",
  "date": "2019-01-02T20:00:00.000Z"
},
"country": "РОССИЙСКАЯ ФЕДЕРАЦИЯ",
"productTypeDesc": "ТАПОЧКИ",
"color": "белый",
"materialDown": "Текстиль",
"productSize": "52",
"materialUpper": "Текстиль",
"materialLining": "Текстиль",
"packageType": "BOX",
"productType": "310000022"
}

```

http response status in case of success: 200 - OK

Request example:

```

curl -X GET
'https://ismp.crpt.ru/api/v3/facade/marked_products/010463003407002921wskg1E4
4Rlqym2406401' -H 'Authorization: Bearer <ТОКЕН>'

```

Ответ:

```

< HTTP/1.1 200
< Server: nginx/1.14.0
< Date: Sun, 27 Jan 2019 09:41:23 GMT
< Content-Type: application/json; charset=UTF-8
< Transfer-Encoding: chunked
< Connection: keep-alive
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1; mode=block
< Cache-Control: no-cache, no-store, max-age=0, must-revalidate
< Pragma: no-cache
< Expires: 0
< X-Frame-Options: DENY
<
* Connection #0 to host ismp.crpt.ru left intact
{
  "cis": "00000000000000000000000000000000",
  "gtin": "00000000000000",
  "tnvd": "6401100000",

```

```

"tnvedGroup": "6401",
"productName": "string",
"ownerName": "string",
"ownerInn": "0000000000",
"producerName": "string",
"producerInn": "0000000000",
"status": "INTRODUCED",
"emissionDate": "2019-01-18T08:14:40.344Z",
"producedDate": "2019-01-18T08:19:33.575Z",
"type": "PRINTED",
"prodOrderType": "OPERATOR",
"lastDocId": "e76a1c45-c8ce-4961-84f1-1df47e5a7d35",
"name": "Белые тапочки, арт. CRPT--99999, размер 52",
"brand": "отсутствует",
"model": "CRPT--99999",
"certDoc": {
  "type": "CONFORMITY_CERT",
  "number": "122",
  "date": "2019-01-02T20:00:00.000Z"
},
"country": "РОССИЙСКАЯ ФЕДЕРАЦИЯ",
"productTypeDesc": "ТАПОЧКИ",
"color": "белый",
"materialDown": "Текстиль",
"productSize": "52",
"materialUpper": "Текстиль",
"materialLining": "Текстиль",
"packageType": "BOX",
"productType": "310000022"
}

```

3.5 Method of receiving of brief information on MC\MC list (publicly available)

URL: /api/v3/facade/cis/cis_list?cis={KM}

Method: GET

URL parameters: cis (array[string]) - MC array

Request header parameters:

- Authorization: Bearer <token>
- Content-Type: application/json

Request example: curl -X GET

"http://10.73.69.69:8095/api/v3/facade/cis/cis_list?cis=1&cis=2" -H 'content-type: application/json' -H 'Authorization: Bearer <TOKEN>'

Response parameters:

Parameter	Type	Description
cis	string	Marking code (MC)
gtin	string	Goods GTIN
productName	string	Product name
emissionDate	number	MC emission date, yyyy-MM-dd'T'HH:mm:ss.SSS'Z format
status	string	MC status: • APPLIED - emitted.Issued • INTRODUCED - released into turnover • RETIRED - dropped out
emissionDate	string	Production date yyyy-MM-dd'T'HH:mm:ss.SSS'Z
ownerName	string	Name of goods owner company
productName	string	Name of goods producing company
statusEx	string	Additional MC status: • EMPTY - additional status is unavailable • WAIT_SHIPMENT - waiting for shipment/acceptance • WAIT_TRANSFER_TO_OWNER - waiting for hand over to the owner (contract production) • WAIT_REMARK - goods item is waiting for remarking or MC is dropped out after application on a goods item
children	array	list of goods identification codes in an aggregate
childrenLength	number	number of goods identification codes in an aggregate

Response:

```
{
  "00000000000000000000000000000000": {
    "cis": "00000000000000000000000000000000",
    "gtin": "04630034070012",
    "producerName": "string",
    "status": "APPLIED",
    "emissionDate": 1550507912681,
    "ownerName": "string",
    "productName": "string",
    "statusEx": "WAIT_SHIPMENT"
  }
}
```

http response status in case of success: 200 - OK

4. Documents

4.1 Method of output of a list of documents previously uploaded to the TT IS

URL: /api/v3/facade/doc/listV2

Method: GET

Request header parameters:

- Authorization: Bearer <token>
- Content-Type: application/json

Request parameters:

Parameter	Type	Necessity	Description
<i>Filtering</i>			
dateFrom	string	-	Emission date “from” is specified in yyyy-MM-dd'T'HH:mm:ss.SSS'Z format, example: 2019-01-01T03:00:00.000Z
dateTo	string	-	Emission date “to” is specified in yyyy-MM-dd'T'HH:mm:ss.SSS'Z format, example 2019-01-10T03:00:00.000Z
number	string	-	Document number
documentStatus	string	-	See the document status in the “ Document status ” table
documentType	Array of string listings	-	Document type. List of available values: UNIVERSAL_TRANSFER_DOCUMENT AGGREGATION_DOCUMENT DISAGGREGATION_DOCUMENT INCLUDE_DOCUMENT

			RECEIPT LP_ADD_GOODS_CSV LP_ADD_GOODS_XML LP_INTRODUCE_GOODS LP_INTRODUCE_OST LP_SHIP_GOODS(28), LP_ACCEPT_GOODS(29), LP_INTRODUCE_GOODS_CSV LP_INTRODUCE_GOODS_XML LP_SHIP_GOODS_CSV LP_SHIP_GOODS_XML LP_ACCEPT_GOODS_XML LK_REMARK LK_REMARK_CSV LK_REMARK_XML LK_RESUME_ACCESS LK_RESUME_ACCESS_XML LK_RECEIPT_XML LK_RECEIPT_CSV INTERNAL_RECEIPT LK_RECEIPT LK_LOAN_RECEIPT_XML LP_GOODS_IMPORT LP_GOODS_IMPORT_CSV
--	--	--	--

			LP_GOODS_IMPORT_XML LK_KM_CANCELLATION LK_KM_CANCELLATION_CSV LK_KM_CANCELLATION_XML LK_APPLIED_KM_CANCELLATION LK_APPLIED_KM_CANCELLATION_CSV LK_APPLIED_KM_CANCELLATION_XML LK_CONTRACT_COMMISSIONING LK_CONTRACT_COMMISSIONING_CSV LK_CONTRACT_COMMISSIONING_XML LK_INDI_COMMISSIONING LK_INDI_COMMISSIONING_CSV LK_INDI_COMMISSIONING_XML AGGREGATION_DOCUMENT_CSV AGGREGATION_DOCUMENT_XML DISAGGREGATION_DOCUMENT_CSV DISAGGREGATION_DOCUMENT_XML REAGGREGATION_DOCUMENT REAGGREGATION_DOCUMENT_CSV REAGGREGATION_DOCUMENT_XML LP_SHIP_RECEIPT LP_SHIP_RECEIPT_CSV LP_SHIP_RECEIPT_XML OST_DESCRIPTION
--	--	--	---

			CROSSBORDER CROSSBORDER_CSV CROSSBORDER_XML
inputFormat	Boolean	-	“true” – for incoming document, “false” – for outgoing document
participantInn	string	-	INN of turnover participant
Pagination/Sorting			
order	String listing	-	Sorting type (ascending/descending). List of available values: • ASC - ascending • DESC - descending
did	string	-	Document number value-"starting points" for records sorting.
orderedColumnValue	string	-	Column value-"starting points" for records sorting.
orderColumn	String listing	-	Column name used for sorting. List of available values: • docDate - document date
pageDir	String listing	-	Selection of direction (forward/backward) for pagination. List of available values: • NEXT - forward • PREV - backward
limit	number	-	Maximum number of records that returns as a response.

Document status:

Name	Description
IN_PROGRESS	under verification
CHECKED_OK	processed (not applied for the “Goods shipment” document)

CHECKED_NOT_OK	verification error
PROCESSING_ERROR	processing error
UNDEFINED	not defined
CANCELLED	shipment cancellation (applied for “Goods shipment” document only)
ACCEPTED	shipment accepted (applied for “Goods shipment” document only)
WAIT_ACCEPTANCE	shipment waiting for acceptance (applied for “Goods shipment” document only)
WAIT_PARTICIPANT_REGISTRATION	shipment for unregistered participant (applied for “Goods shipment” document only)

Response parameters:

Parameter	Type	Description
number	string	Document number
docDate	string	Document date, yyyy-MM-dd'T'HH:mm:ss.SSS'Z format
receivedAt	string	Unix-time of document receiving
type	string	Document type
status	string	See the document status in the “ Document status ” table
senderName	string	Name of a document sender

Response:

```
{
  "results": [
    {
      "number": "string",
      "docDate": "2019-01-18T06:45:35.630Z",
      "receivedAt": 1547793935630,
      "type": "LP_INTRODUCE_GOODS",
      "status": "CHECKED_OK",
      "senderName": "string",
    }
  ]
}
```

http response status in case of success: 200 - OK

Request example:

```
curl
'https://ismp.crpt.ru/api/v3/facade/doc/listV2?limit=10&order=DESC&orderColumn=docDate&did=623136d3-7a9b-40c9-8ce3-8091e41f83aa&orderedColumnValue=2019-01-28T09:30:40.136Z&pageDir=NEXT' -H 'content-type: application/json' -H
'Authorization: Bearer <TOKEH>'
```

```
OTBet:
< HTTP/1.1 200
< Server: nginx/1.14.0
< Date: Sun, 27 Jan 2019 18:10:33 GMT
< Content-Type: application/json; charset=UTF-8
< Transfer-Encoding: chunked
< Connection: keep-alive
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1; mode=block
< Cache-Control: no-cache, no-store, max-age=0, must-revalidate
< Pragma: no-cache
< Expires: 0
< X-Frame-Options: DENY
<

* Connection #0 to host ismp.crpt.ru left intact
{
  "results": [
    {
      "number": "string",
      "docDate": "2019-01-18T06:45:35.630Z",
      "receivedAt": 1547793935630,
      "type": "LP_INTRODUCE_GOODS",
      "status": "CHECKED_OK",
      "senderName": "string",
    }
  ]
}
```

4.2 Method of content output of documents previously uploaded to the TT IS

URL: /api/v3/facade/doc/{docId}/body

Method: GET

URL parameters: docId - document number

Request header parameters:

- Authorization: Bearer <token>
- Content-Type: application/json

Response parameters:

Parameter	Type	Description
number	string	Document number
docDate	string	Document date, yyyy-MM-dd'T'HH:mm:ss.SSS'Z format
receivedAt	string	Unix-time of document receiving
type	string	Document type
status	string	See the document status in the “ Document status ” table
senderName	string	Name of a document sender
body	string	Document content in JSON format. This JSON can contain a set of fields special for particular document.
content	string	Original document content (if the document was added to the system by CSV/XML-file loading, the field will include the content of this CSV/XML-file).
input	Boolean	Criteria of incoming/outgoing document (“true” value for incoming document, “false” for outgoing document)

Response:

```
{
  "number": <номер документа>,
  "docDate": <дата создания >, пример "2019-01-18T08:19:33.575Z",
  "type": <тип документа>, пример "LP_INTRODUCE_GOODS",
  "status": <статус проверки>, пример "CHECKED_OK",
  "senderName": <наименование отправителя>,
  "body": <тело документа в json>,
  "content": <тело документа в исходном формате>
}
```

http response status in case of success: 200 - OK

Request example:

```
curl -X GET 'https://ismp.crpt.ru/api/v3/facade/doc/e76a1c45-c8ce-4961-84f1-1df47e5a7d35/body' -H 'content-type: application/json' -H 'Authorization: Bearer <TOKEH>'
```

Ответ:

```
< HTTP/1.1 200
< Server: nginx/1.14.0
< Date: Sun, 27 Jan 2019 18:10:33 GMT
< Content-Type: application/json; charset=UTF-8
< Transfer-Encoding: chunked
< Connection: keep-alive
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1; mode=block
< Cache-Control: no-cache, no-store, max-age=0, must-revalidate
```

< Pragma: no-cache
< Expires: 0
< X-Frame-Options: DENY
<

* Connection #0 to host ismp.crpt.ru left intact

```
{
  "number": "e76a1c45-c8ce-4961-84f1-1df47e5a7d35",
  "docDate": "2019-01-18T08:19:33.575Z",
  "type": "LP_INTRODUCE_GOODS",
  "status": "CHECKED_OK",
  "senderName": "string",
  "downloadStatus": "CHECKED_OK",
  "body": {
    "reg_date": "2019-01-18T08:19:31.967Z",
    "document_description": {
      "participant_inn": "0000000000",
      "production_date": "2019-01-17T20:00:00.000Z",
      "production_type": "OWN_PRODUCTION",
      "producer_inn": "0000000000",
      "owner_inn": "0000000000"
    },
    "doc_type": "Promotion_Inform_Selfmade",
    "doc_id": "string",
    "products": [
      {
        "uitu_code": "",
        "uit_code": "00000000000000000000000000000000",
        "tnved_code": "6401100000",
        "producer_inn": "0000000000",
        "owner_inn": "0000000000",
        "certificate_document": "CONFORMITY_CERT",
        "certificate_document_number": "122",
        "certificate_document_date": "2019-01-02T20:00:00.000Z"
      }
    ],
    "products_list": [
      {
        "product_gender": "UNISEX",
        "tnved_code_2": "64",
        "release_method": "PRODUCED_IN_RF",
        "gtin": "02900000000766",
        "id": 4431097
      }
    ]
  },
  "content":
  "{\\"doc_id\\":\\"string\\",\\"doc_type\\":\\"Promotion_Inform_Selfmade\\",\\"reg_date\\":\\"2019-01-18T08:19:31.967Z\\",\\"document_description\\":{\\"participant_inn\\":\\"0000000000\\",\\"production_date\\":\\"2019-01-17T20:00:00.000Z\\",\\"production_type\\":\\"OWN_PRODUCTION\\",\\"producer_inn\\":\\"0000000000\\",\\"owner_inn\\":\\"0000000000\\"},\\"products\\":[{\\"uitu_code\\":\\"\\",\\"uit_code\\":\\"00000000000000000000000000000000\\",\\"tnved_code\\":\\"6401100000\\",\\"producer_inn\\":\\"7731369928\\",\\"owner_inn\\":\\"7731369928\\",\\"certificate_document\\":\\"CONFORMITY_CERT\\",\\"certificate_document_number\\":\\"122\\",\\"certificate_document_date\\":\\"2019-01-02T20:00:00.000Z\\"}]]}"
```

```
}
```

4.3 Unified method of documents creation

URL: /api/v3/lk/documents/create

Method: POST

Request header parameters:

- Authorization: Bearer <token>
- Content-Type: application/json

URL request parameters:

Parameter	Type	Necessity	Description
pg	string	+	Name of the goods group (see the list of available goods groups in the “List of supported goods groups” appendix).

Request body parameters:

Parameter	Type	Necessity	Description
product_document	string	+	Document on goods release into turnover encoded in base64. The document is described in “Appendix. Document on release into turnover” clause.
document_format	string	+	Document format on goods release into turnover: • MANUAL - json • CSV - csv • XML - xml
type	string	+	Document type (see the “Types of documents” table)
signature	string	+	Detached EQES encoded in base64. The document shall be signed prior to its conversion to base64

Request body example:

```
{  
  "product_document": "<Документ в base64>",  
}
```

```

    "document_format": "<Формат документа>",
    "type": "<Тип>",
    "signature": "<Открепленная подпись в base64>"
}

```

Document types:

Type	Name	Format
LP_INTRODUCE_GOODS	Release into turnover. Production in the Russian Federation.	json
LP_INTRODUCE_GOODS_CSV	Release into turnover. Production in the Russian Federation.	cdv
LP_INTRODUCE_GOODS_XML	Release into turnover. Production in the Russian Federation.	xml
LP_GOODS_IMPORT	Release into turnover. Production outside EEU.	json
LP_GOODS_IMPORT_CSV	Release into turnover. Production outside EEU.	csv
LP_GOODS_IMPORT_XML	Release into turnover. Production outside EEU.	xml
LK_CONTRACT_COMMISSIONING	Goods release into turnover. Contract production in the Russian Federation.	json
LK_CONTRACT_COMMISSIONING_CSV	Goods release into turnover. Contract production in the Russian Federation.	csv
LK_CONTRACT_COMMISSIONING_XML	Goods release into turnover. Contract production in the Russian Federation.	xml
LK_INDI_COMMISSIONING	Goods release into turnover. Received from physical entities.	json
LK_INDI_COMMISSIONING_CSV	Goods release into turnover. Received from physical entities.	csv
LK_INDI_COMMISSIONING_XML	Goods release into turnover. Received from physical entities.	xml
LP_INTRODUCE_OST	Release into turnover. Marking of remaining amount	json
LP_INTRODUCE_OST_CSV	Release into turnover. Marking of remaining amount	csv
LP_INTRODUCE_OST_XML	Release into turnover. Marking of	xml

	remaining amount	
CROSSBORDER	Release into turnover. Cross-border trade	json
CROSSBORDER_CSV	Release into turnover. Cross-border trade	csv
CROSSBORDER_XML	Release into turnover. Cross-border trade	xml
LP_SHIP_GOODS	Shipment	json
LP_SHIP_GOODS_CSV	Shipment	csv
LP_SHIP_GOODS_XML	Shipment	xml
LP_CANCEL_SHIPMENT	Shipment cancellation	json
LP_CANCEL_SHIPMENT_CSV	Shipment cancellation	csv
LP_CANCEL_SHIPMENT_XML	Shipment cancellation	xml
LP_ACCEPT_GOODS	Acceptance	json
LP_ACCEPT_GOODS_XML	Acceptance	xml
LK_REMARK	Remarking	json
LK_REMARK_CSV	Remarking	csv
LK_REMARK_XML	Remarking	xml
LK_KM_CANCELLATION	Drop out of non-applied MC	json
LK_KM_CANCELLATION_CSV	Drop out of non-applied MC	csv
LK_KM_CANCELLATION_XML	Drop out of non-applied MC	xml
LK_APPLIED_KM_CANCELLATION	Drop out of applied MC	json
LK_APPLIED_KM_CANCELLATION_CSV	Drop out of applied MC	csv
LK_APPLIED_KM_CANCELLATION_XML	Drop out of applied MC	xml
AGGREGATION_DOCUMENT	Aggregation	json
AGGREGATION_DOCUMENT_CSV	Aggregation	csv
AGGREGATION_DOCUMENT_XML	Aggregation	xml
DISAGGREGATION_DOCUMENT	Disaggregation	json
DISAGGREGATION_DOCUMENT_CSV	Disaggregation	csv

DISAGGREGATION_DOCUMENT_XML	Disaggregation	xml
REAGGREGATION_DOCUMENT	Re-aggregation	json
REAGGREGATION_DOCUMENT_CSV	Re-aggregation	csv
REAGGREGATION_DOCUMENT_XML	Re-aggregation	xml
OST_DESCRIPTION	Remaining goods description	json
OST_DESCRIPTION_CSV	Remaining goods description	csv
OST_DESCRIPTION_XML	Remaining goods description	xml

Response parameters:

Parameter	Type	Description
GUID	string	Unique identifier of a document in the TT IS

Response:

<GUID>

http response status in case of success: 200 - OK

Request example:

```
curl 'https://ismp.crpt.ru/api/v3/lk/documents/send' -H 'content-type: application/json' -H 'Authorization: Bearer <TOKEH>' --data-binary '{ "product_document": "<Документ в base64>", "document_format": "MANUAL", "type": "LP_INTRODUCE_GOODS", "signature": "<Открепленная подпись в base64>" }'
```

Ответ:

```
< HTTP/1.1 200
< Server: nginx/1.14.0
< Date: Sun, 27 Jan 2019 18:10:33 GMT
< Content-Type: application/json; charset=UTF-8
< Transfer-Encoding: chunked
< Connection: keep-alive
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1; mode=block
< Cache-Control: no-cache, no-store, max-age=0, must-revalidate
< Pragma: no-cache
< Expires: 0
< X-Frame-Options: DENY
<
```

```
* Connection #0 to host ismp.crpt.ru left intact
9abd3d41-76bc-4542-a88e-b1f7be8130b5
```

4.4 Methods of documents creation

URL of a request on creation of particular document type:

Document type	URL of document creation method
Release into turnover. Production in the Russian Federation.	/api/v3/lk/documents/send
Release into turnover. Production outside EEU	/api/v3/lk/import/send
Release into turnover. Contract production in the Russian Federation	/api/v3/lk/documents/commissioning/contract/create
Release into turnover. Received from physical entities	/api/v3/lk/documents/commissioning/indi/create
Release into turnover. Cross-border trade	
Release into turnover. Marking of remaining amount	
Goods shipment	/api/v3/lk/documents/shipment/create
Goods shipment cancellation	/api/v3/lk/documents/shipment/cancel
Goods acceptance	/api/v3/lk/documents/acceptance/create
Goods withdrawal from turnover at contract implementation for the final consumer	/api/v3/lk/receipt/send
Drop out of MC non-applied on goods with/without IM	/api/v3/lk/documents/km/cancellation/create
Drop out of MC applied on goods with IM (without goods withdrawal from turnover)	/api/v3/lk/documents/km/cancellation/applied/create
Goods remarking	/api/v3/lk/remarking/send
Aggregation	/api/v3/lk/documents/aggregation/create
Disaggregation	/api/v3/lk/documents/disaggregation/create
Re-aggregation	/api/v3/lk/documents/reaggregation/create
Remaining goods description	

Method: POST

Request header parameters:

- Authorization: Bearer <token>
- Content-Type: application/json

URL request parameters:

Parameter	Type	Necessity	Description
pg	string	+	Name of the goods group (see the list of available goods groups in the “List of supported goods groups” appendix).

Request body parameters:

Parameter	Type	Necessity	Description
product_document	string	+	Document on goods release into turnover encoded in base64. The document is described in “Appendix. Document on release into turnover” clause.
document_format	string	+	Document format on goods release into turnover: • MANUAL - json • CSV - csv • XML - xml
signature	string	+	Detached EQES encoded in base64. The document shall be signed prior to its conversion to base64

Request body example:

```
{  
  "product_document": "<Документ в base64>",  
  "document_format": "<Формат документа>",  
  "signature": "<Открепленная подпись в base64>"  
}
```

Response parameters:

Parameter	Type	Description
GUID	string	Unique identifier of a document in the TT IS

Response:

<GUID>

http response status in case of success: 200 - OK

Request example:

```
curl 'https://ismp.crpt.ru/api/v3/lk/documents/send' -H 'content-type: application/json' -H 'Authorization: Bearer <TOKEH>' --data-binary '{ "product_document": "<Документ в base64>", "document_format": "MANUAL", "signature": "<Открепленная подпись в base64>"}'
```

ОТВЕТ:

```
< HTTP/1.1 200
< Server: nginx/1.14.0
< Date: Sun, 27 Jan 2019 18:10:33 GMT
< Content-Type: application/json; charset=UTF-8
< Transfer-Encoding: chunked
< Connection: keep-alive
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1; mode=block
< Cache-Control: no-cache, no-store, max-age=0, must-revalidate
< Pragma: no-cache
< Expires: 0
< X-Frame-Options: DENY
<
```

```
* Connection #0 to host ismp.crpt.ru left intact
9abd3d41-76bc-4542-a88e-b1f7be8130b5
```

4.5 Method of draft document creation

URL: /api/v3/lk/documents/draft

Method: POST

Request header parameters:

- Authorization: Bearer <token>
- Content-Type: application/json

Request body parameters:

Parameter	Type	Necessity	Description
date_saved	string	+	Saving date, yyyy-MM-dd'T'HH:mm:ss.SSS'Z format
document_body	string	+	Document body encoded in base64.
document_type	string	+	Document type

organisation_inn	string	+	Organization INN
registration_date	string	+	Creation date, <i>yyyy-MM-dd'T'HH:mm:ss.SSS'Z</i> format
user_id	string	+	User identifier

Body request:

```
{
  "date_saved": "2019-02-13T07:59:05.785Z",
  "document_body": "string",
  "document_type": "UNKNOWN",
  "organisation_inn": "string",
  "registration_date": "2019-02-13T07:59:05.785Z",
  "user_id": 0
}
```

Response parameters:

Parameter	Type	Description
date_saved	string	Saving date, <i>yyyy-MM-dd'T'HH:mm:ss.SSS'Z</i> format
document_body	string	Document date, <i>yyyy-MM-dd'T'HH:mm:ss.SSS'Z</i> format
document_type	string	Document type
document_uuid	string	Document identifier
draft_uuid	string	Draft document identifier
errors	string	Error description, if any
organisation_inn	string	Organization INN
registration_date	string	Creation date, <i>yyyy-MM-dd'T'HH:mm:ss.SSS'Z</i> format
user_id	string	User identifier

Response:

```
{
  "date_saved": "2019-02-13T08:18:50.932Z",
  "document_body": "string",
  "document_type": "UNKNOWN",
  "document_uuid": "string",
  "draft_uuid": "string",
  "errors": "string",
  "organisation_inn": "string",
  "registration_date": "2019-02-13T08:18:50.932Z",
  "user_id": 0
}
```

http response status in case of success: 200 - OK

4.6 Method of draft document receiving

URL: /api/v3/lk/documents/draft/{id}

Method: GET

Request header parameters:

- Authorization: Bearer <token>
- Content-Type: application/json

Response parameters:

Parameter	Type	Description
date_saved	string	Saving date, yyyy-MM-dd'T'HH:mm:ss.SSS'Z format
document_body	string	Document date, yyyy-MM-dd'T'HH:mm:ss.SSS'Z format
document_type	string	Document type
document_uuid	string	Document identifier
draft_uuid	string	Draft document identifier
errors	string	Error description, if any
organisation_inn	string	Organization INN
registration_date	string	Creation date, yyyy-MM-dd'T'HH:mm:ss.SSS'Z format
user_id	string	User identifier

Response:

```
{
  "date_saved": "2019-02-13T10:28:36.520Z",
  "document_body": "string",
  "document_type": "UNKNOWN",
  "document_uuid": "string",
  "draft_uuid": "string",
  "errors": "string",
  "organisation_inn": "string",
  "registration_date": "2019-02-13T10:28:36.521Z",
  "user_id": 0
}
```

http response status in case of success: 200 - OK

http response status in case of "Draft not found" error: 204

4.7 Method of draft document update

URL: /api/v3/lk/documents/draft/{id}

Method: PUT

URL parameters: id - draft identifier

Request header parameters:

- Authorization: Bearer <token>
- Content-Type: application/json

Request body parameters:

Parameter	Type	Necessity	Description
date_saved	string	+	Saving date, yyyy-MM-dd'T'HH:mm:ss.SSS'Z format
document_body	string	+	Document body encoded in base64.
document_type	string	+	Document type
organisation_inn	string	+	Organization INN
registration_date	string	+	Creation date, yyyy-MM-dd'T'HH:mm:ss.SSS'Z format
user_id	string	+	User identifier

Body request:

```
{
  "date_saved": "2019-02-13T07:59:05.785Z",
  "document_body": "string",
  "document_type": "UNKNOWN",
  "organisation_inn": "string",
  "registration_date": "2019-02-13T07:59:05.785Z",
  "user_id": 0
}
```

Response:

```
{
  "date_saved": "2019-02-13T10:58:47.990Z",
  "document_body": "string",
  "document_type": "UNKNOWN",
  "document_uuid": "string",
  "draft_uuid": "string",
  "errors": "string",
}
```

```
"organisation_inn": "string",
"registration_date": "2019-02-13T10:58:47.990Z",
"user_id": 0
}
```

http response status in case of success: 200 - OK

4.8 Method of draft document deletion

URL: /api/v3/lk/documents/draft/{id}

Method: DELETE

URL parameters: id - draft identifier

Request header parameters:

- Authorization: Bearer <token>

http response status in case of success: 200 - OK

4.9 Method of draft document searching and filtering

URL: /api/v3/lk/documents/draft/search

Method: GET

Request header parameters:

- Authorization: Bearer <token>

Request parameters:

Parameter	Type	Necessity	Description
<i>Filtering</i>			
id	string	-	Draft identifier
registerDateStart	string	-	Registration date “from” is specified in yyyy-MM-dd'T'HH:mm:ss.SSS'Z format, example: 2019-01-01T03:00:00.000Z
registerDateEnd	string	-	Registration date “to” is specified in yyyy-MM-dd'T'HH:mm:ss.SSS'Z format, example: 2019-01-01T03:00:00.000Z

dateSavedStart	string	-	Date of last saving “from” is specified in yyyy-MM-dd'T'HH:mm:ss.SSS'Z format, example: 2019-01-01T03:00:00.000Z
dateSavedEnd	string	-	Date of last saving “to” is specified in yyyy-MM-dd'T'HH:mm:ss.SSS'Z format, example: 2019-01-01T03:00:00.000Z
userId	string	-	User identifier
organisationInn	string	-	Organization INN
documentType	string	-	Document type
orderColumn	string	-	Sorting field
orderDirection	string	-	Sorting type: ASC/DESC
limit	number	-	Section size
page	number	-	Section number
onDelete	boolean	-	Deletion check box: true/false.

Request example:

<http://shoes.crpt.ru/api/v3/lk/documents/draft/search?limit=10&orderDirection=ASC&page=2&userId=1>

Response:

```
[
  {
    "draft_uuid": "0b35d630-47e6-45df-b0e0-12fe04540042",
    "registration_date": "2019-02-13T21:53:47.708Z",
    "date_saved": "2019-02-13T21:53:47.708Z",
    "user_id": 2,
    "organisation_inn": "0000000000",
    "document_type": "UNKNOWN",
    "document_uuid": "string",
    "errors": "string"
  }
]
```

http response status in case of success: 200 - OK

5. Goods guide GS1/National catalogue

5.1 Method of the GTIN list output for a turnover participant by INN

URL: /api/v3/product/search

Method: GET

Request header parameters:

- Authorization: Bearer <token>

URL parameters:

Parameter	Type	Necessity	Description
limit	string	-	Maximum number of MC orders in a response.
page	number	-	Response page number, starting value 0.
pg	string	-	Name of the goods group (see the list of available goods groups in the “List of supported goods groups” appendix)
gtin	string	-	Goods code

Example of response parameters (List of returned parameters can be different depending on code goods group):

Parameter	Type	Description
id	number	Goods unique number
name	string	Goods name
gtin	string	Goods GTIN
brand	string	Brand name
packageType	string	Packing type:
innerUnitCount	number	Number of goods copies
model	string	Model of a producer
publicationDate	number	Posting date. Unix-time format in milliseconds
inn	string	INN of a Russian producer
country	string	Country of origin
productTypeDesc	string	Label name

color	string	Colour
materialDown	string	Bottom material
materialUpper	string	Top material
materialLining	string	Lining material
productType	string	Product type
productGroupId	number	Goods group identifier in compliance with the “Goods groups codes” table
productSize	string	Dimension
goodSignedFlag	boolean	Check box of GTIN card signing in the National catalogue. If GTIN is received from GS1, “true” value always returns
goodMarkFlag	boolean	Check box showing if the first attributes layer is filled in in the National catalogue. GTINs that checked “false” do not return in responses and are not saved in the DB. If GTIN is received from GS1, “true” value always returns
goodTurnFlag	boolean	Check box showing if the second attributes layer is filled in in the National catalogue. If GTIN is received from GS1, “true” value always returns

Response:

```
{
  "errorCode": <Код ошибки>,
  "results": <Информация по товарам> [
    {
      "id": <ID товара>, пример 30056,
      "name": <Наименование товара>
      "gtin": "<GTIN товара>, пример "04690257113712",
      "brand": <Наименование бренда>
      "packageType": <Тип упаковки>, пример "BOX",
      "innerUnitCount": <Число экземпляров товара>, пример 1,
      "model": <Модель производителя>, пример "001079-11",
      "publicationDate": <Дата публикации>, пример 1537304400000,
      "inn": <ИНН Российского производителя>
      "country": <Страна производства>, пример "КИТАЙ",
      "productTypeDesc": <Наименование на этикетке>, пример "ПИНЕТКИ",
      "color": <Цвет>, пример "белый",
      "materialDown": <Материал низа>, пример "Текстиль",
      "materialUpper": <Материал верха>, пример "Текстиль",
      "materialLining": <Материал подкладки>, пример "Текстиль",
      "productType": <Тип продукта>, пример "310000013",
      "productGroupId": <ID товарной группы>, пример 2,
      "goodSignedFlag": <Флаг наличия подписи в НК>, пример true,
      "goodMarkFlag": <Флаг заполненности атрибутов первого слоя в НК>,
    пример true,
  ]
}
```

```
        "goodTurnFlag": <Флаг заполненности атрибутов второго слоя в НК>,
пример true
    }
    ],
    "total": <Количество товаров>
}
```

HTTP-response status in case of success: 200 - OK

Request example:

```
curl -X GET
'https://ismp.crpt.ru/api/v3/product/search?limit=10&offset=0&page=0&pg=shoes
' -H 'Authorization: Bearer <ТОКЕН>'
```

Ответ:

```
< HTTP/1.1 200
< Server: nginx/1.14.0
< Date: Sun, 27 Jan 2019 18:10:33 GMT
< Content-Type: application/json; charset=UTF-8
< Transfer-Encoding: chunked
< Connection: keep-alive
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1; mode=block
< Cache-Control: no-cache, no-store, max-age=0, must-revalidate
< Pragma: no-cache
< Expires: 0
< X-Frame-Options: DENY
```

* Connection #0 to host ismp.crpt.ru left intact

```
{
  "results": [
    {
      "id": id,
      "name": "string",
      "gtin": "123456789101",
      "brand": "КОТОФЕЙ",
      "packageType": "BOX",
      "innerUnitCount": 1,
      "model": "001079-11",
      "publicationDate": 1537304400000,
      "inn": "5011017647",
      "country": "КИТАЙ",
      "productTypeDesc": "string",
      "color": "белый",
      "materialDown": "Текстиль",
      "materialUpper": "Текстиль",
      "materialLining": "Текстиль",
      "productType": "310000013",
      "productGroupId": 2,
      "goodSignedFlag": true,
      "goodMarkFlag": true,
      "goodTurnFlag": true
    },
    {
      "id": id,
```

```

        "name": "string",
        "gtin": "04690257114078",
        "brand": "КОТОФЕЙ",
        "packageType": "BOX",
        "innerUnitCount": 1,
        "model": "001079-12",
        "publicationDate": 1537304400000,
        "inn": "5011017647",
        "country": "КИТАЙ",
        "productTypeDesc": "ПИНЕТКИ",
        "color": "розовый",
        "materialDown": "Текстиль",
        "materialUpper": "Текстиль",
        "materialLining": "Текстиль",
        "productType": "310000013",
        "productGroupId": 2,
        "goodSignedFlag": true,
        "goodMarkFlag": true,
        "goodTurnFlag": true
    },
    {
        "id": 30059,
        "name": "string",
        "gtin": "1234567891012",
        "brand": "string",
        "packageType": "BOX",
        "innerUnitCount": 1,
        "model": "001079-12",
        "publicationDate": 1537304400000,
        "inn": "5011017647",
        "country": "КИТАЙ",
        "productTypeDesc": "ПИНЕТКИ",
        "color": "розовый",
        "materialDown": "Текстиль",
        "materialUpper": "Текстиль",
        "materialLining": "Текстиль",
        "productType": "310000013",
        "productGroupId": 2,
        "goodSignedFlag": true,
        "goodMarkFlag": true,
        "goodTurnFlag": true
    }
],
"total": 11210,
"errorCode": null
}

```

5.2 Method of information receiving on a goods item by GTIN

URL: api/v3/product/info

Method: GET

Request header parameters:

- Authorization: Bearer <token>

URL parameters: gtins – GTIN list.

Example of response parameters (List of returned parameters can be different depending on code goods group):

Parameter	Type	Description
id	number	Goods unique number
name	string	Goods name
gtin	string	Goods GTIN
brand	string	Brand name
packageType	string	Packing type:
innerUnitCount	number	Number of goods copies
model	string	Model of a producer
publicationDate	number	Posting date. Unix-time format in milliseconds
inn	string	INN of a Russian producer
country	string	Country of origin
productTypeDesc	string	Label name
color	string	Colour
materialDown	string	Bottom material
materialUpper	string	Top material
materialLining	string	Lining material
productType	string	Product type
productGroupId	number	Goods group identifier in compliance with the “Supported goods groups list” appendix
goodSignedFlag	boolean	Check box of GTIN card signing in the National catalogue. If GTIN is received from GS1, true value always returns
goodMarkFlag	boolean	Check box showing if the first attributes layer is filled in in the National catalogue. GTINs that checked “false” do not return in responses and are not saved in the DB. If GTIN is received from GS1, “true” value always returns

goodTurnFlag	boolean	Check box showing if the second attributes layer is filled in in the National catalogue. If GTIN is received from GS1, "true" value always returns
--------------	---------	--

HTTP-response status in case of success: 200 - OK

Request example:

```
curl -X GET
'https://ismp.crpt.ru/api/v3/product/api/v3/product/info?gtins=04690257113712
,04690257113712' -H 'Authorization: Bearer <TOKEH>'
```

Ответ:

```
< HTTP/1.1 200
< Server: nginx/1.14.0
< Date: Sun, 27 Jan 2019 18:10:33 GMT
< Content-Type: application/json; charset=UTF-8
< Transfer-Encoding: chunked
< Connection: keep-alive
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1; mode=block
< Cache-Control: no-cache, no-store, max-age=0, must-revalidate
< Pragma: no-cache
< Expires: 0
< X-Frame-Options: DENY
```

* Connection #0 to host ismp.crpt.ru left intact

```
{
  "errorCode": <Код ошибки>,
  "results": <Информация по товарам> [
    {
      "id": <ID товара>, пример 30056,
      "name": <Наименование товара>
      "gtin": "<GTIN товара>, пример
      "brand": <Наименование бренда>
      "packageType": <Тип упаковки>, пример "BOX",
      "innerUnitCount": <Число экземпляров товара>, пример 1,
      "model": <Модель производителя>, пример "001079-11",
      "publicationDate": <Дата публикации>, пример 1537304400000,
      "inn": <ИНН Российского производителя>
      "country": <Страна производства>, пример "КИТАЙ",
      "productTypeDesc": <Наименование на этикетке>, пример "ПИНЕТКИ",
      "color": <Цвет>, пример "белый",
      "materialDown": <Материал низа>, пример "Текстиль",
      "materialUpper": <Материал верха>, пример "Текстиль",
      "materialLining": <Материал подкладки>, пример "Текстиль",
      "productType": <Тип продукта>, пример "310000013",
      "productGroupId": <ID товарной группы>, пример 2,
      "goodSignedFlag": <Флаг наличия подписи в НК>, пример true,
      "goodMarkFlag": <Флаг заполненности атрибутов первого слоя в НК>,
      пример true,
      "goodTurnFlag": <Флаг заполненности атрибутов второго слоя в НК>,
      пример true
    }
  ],
```

```
"total": <Количество товаров>
}
```

6. Guide of TNVED codes

6.1 Method of output of 10-character TNVED codes list

URL: /api/v3/facade/tnved/search

Method: GET

Request header parameters:

- Authorization: Bearer <token>

URL parameters:

Parameter	Type	Necessity	Description
prefix	string	-	Array of 4-character TNVED codes used as prefixes of 10-character TNVED codes in a response.
limit	string	-	Maximum number of MC orders in a response.
offset	string	-	Response page number, starting value 0.

Response parameters:

Parameter	Type	Description
records - arrays including a list of 10-character TNVED codes meeting the request conditions		
code	string	10-character TNVED code
description	string	Text description of TNVED code

Response:

```
{
  "records": [
    {
      "code": <10-ти значный код ТН ВЭД>, пример "6401100000",
      "description": <Описание группы товаров>
    },
    {
      "code": <10-ти значный код ТН ВЭД>, пример "6401100000",
      "description": <Описание группы товаров>
    }
  ]
}
```

```
],
"total": 71
}
```

http response status in case of success: 200 - OK

Request example:

```
curl -X GET
'https://ismp.crpt.ru/api/v3/facade/tnved/search?limit=10&offset=0&page=0&pre
fix=6401,6402,6403,6404,6405' -H 'Authorization: Bearer <TOKEN>'
```

```
Ответ:
< HTTP/1.1 200
< Server: nginx/1.14.0
< Date: Sun, 27 Jan 2019 18:10:33 GMT
< Content-Type: application/json; charset=UTF-8
< Transfer-Encoding: chunked
< Connection: keep-alive
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1; mode=block
< Cache-Control: no-cache, no-store, max-age=0, must-revalidate
< Pragma: no-cache
< Expires: 0
< X-Frame-Options: DENY

* Connection #0 to host ismp.crpt.ru left intact
{
  "records": [
    {
      "code": "6401100000",
      "description": "ОБУВЬ, ГЕТРЫ И АНАЛОГИЧНЫЕ ИЗДЕЛИЯ; ИХ ДЕТАЛИ.
ВОДОНЕПРОНИЦАЕМАЯ ОБУВЬ С ПОДОШВОЙ И С ВЕРХОМ ИЗ РЕЗИНЫ ИЛИ ПЛАСТМАССЫ, ВЕРХ
КОТОРОЙ НЕ КРЕПИТСЯ К ПОДОШВЕ И НЕ СОЕДИНЯЕТСЯ С НЕЙ НИ НИТОЧНЫМ, НИ
ШПИЛЕЧНЫМ, НИ ГВОЗДЕВЫМ, НИ ВИНТОВЫМ, НИ ЗАКЛЕПОЧНЫМ, НИ КАКИМ-ЛИБО ДРУГИМ
АНАЛОГИЧНЫМ СПОСОБОМ. - обувь с защитным металлическим подноском"
    },
    {
      "code": "6401921000",
      "description": "ОБУВЬ, ГЕТРЫ И АНАЛОГИЧНЫЕ ИЗДЕЛИЯ; ИХ ДЕТАЛИ.
ВОДОНЕПРОНИЦАЕМАЯ ОБУВЬ С ПОДОШВОЙ И С ВЕРХОМ ИЗ РЕЗИНЫ ИЛИ ПЛАСТМАССЫ, ВЕРХ
КОТОРОЙ НЕ КРЕПИТСЯ К ПОДОШВЕ И НЕ СОЕДИНЯЕТСЯ С НЕЙ НИ НИТОЧНЫМ, НИ
ШПИЛЕЧНЫМ, НИ ГВОЗДЕВЫМ, НИ ВИНТОВЫМ, НИ ЗАКЛЕПОЧНЫМ, НИ КАКИМ-ЛИБО ДРУГИМ
АНАЛОГИЧНЫМ СПОСОБОМ. - - - с верхом из резины"
    }
  ],
  "total": 71
}
```

7. Appendix.

The actual XSD schemes, examples of XML, CSV are specified in 'Help' section <https://shoes.crpt.ru/help/list>.

8. Appendix. List of supported goods groups

Name (code)	Description
clothes	Items of clothing, bed linen, table linen, bath and kitchen linen
shoes	Footwear
perfumery	Perfume and eau de toilette
tires	Tyres and new pneumatic rubber tire casings
electronics	Photo cameras (except cine camera), flash lights and flash bulbs
milk	Dairy products
bicycle	Bicycles and bicycle frames
wheelchairs	Wheelchairs

9. Appendix. Document on release into turnover. Production in the Russian Federation.

9.1 JSON format

Example:

```
{
  "document_description" : {
    "participant_inn": "0000000000",
    "production_date": "2012-11-22T20:13:10.002Z",
    "producer_inn": "0000000000",
    "owner_inn": "0000000000",
    "production_type": "OWN_PRODUCTION"
  },
  "doc_type": "Promotion_Inform_Selfmade",
  "products": [
    {
```

```

    "production_date": "2012-11-22T20:13:10.002Z",
    "uit_code": "00000000000000000000000000000000",
    "uitu_code": "00000000000000000000",
    "tnved_code": "0123456789",
    "producer_inn": "0000000000",
    "owner_inn": "0000000000",
    "certificate_document": "CONFORMITY_CERTIFICATE",
    "certificate_document_number": "FFFFFF",
    "certificate_document_date": "2014-11-22T20:13:10.002Z"
  }
}

```

Attribute content:

Name	Description	Type	Mandatory
participant_inn	INN of a turnover participant	string	yes
producer_inn	INN of a producer	string	yes
production_type	method of production Possible values of OWN_PRODUCTION - own production	string	yes
production_date	production date	string	yes
uit_code	Goods unique identifier (GUI)	string	Mandatory if uitu is not specified
uitu_code	shipment packing unique identifier	string	Mandatory if uit is not specified
tnved_code	10-character TNVED code	string	yes
certificate_document	type of mandatory certification document. Possible value CONFORMITY_CERTIFICATE- conformity certificate, CONFORMITY_DECLARATION - conformity declaration.	string	yes
certificate_document_number	number of mandatory certification document	string	yes

certificate_document_date	date of mandatory certification document	string	yes
---------------------------	--	--------	-----

10. Appendix. Document on release into turnover. Production outside EEU

10.1 JSON format

Example:

```
{
  "doc_type": "GOODSIMPORT",
  "document_description": {
    "participant_inn": "0000000000",
    "declaration_date": "2019-12-31T21:00:00.000Z",
    "declaration_number": "decl num",
    "customs_code": "customs code",
    "decision_code": 1
  },
  "products": [
    {
      "uit_code": "00000000000000000000000000000000",
      "uitu_code": "000000000000000000",
      "tnved_code": "0123456789",
      "certificate_document": "CONFORMITY_DECLARATION",
      "certificate_document_number": "FFFFFF",
      "certificate_document_date": "2014-11-22T21:00:00.000Z",
    }
  ]
}
```

Attribute content:

Name	Description	Type	Mandatory
participant_inn	INN of a turnover participant.	string	yes
declaration_date	Goods declaration date	string	yes
declaration_number	Goods declaration number	string	yes
customs_code	Customs body code	string	yes
decision_code	Decision code	string	yes

uit_code	Goods unique identifier	string	Mandatory if uitu is not specified
uitu_code	shipment packing unique identifier	string	Mandatory if uit is not specified
tnved_code	10-character TNVED code	string	yes
certificate_document	type of mandatory certification document. Possible value CONFORMITY_CERTIFICATE- conformity certificate, CONFORMITY_DECLARATION - conformity declaration.	string	yes
certificate_document_number	number of mandatory certification document	string	yes
certificate_document_date	date of mandatory certification document	string	yes

11. Appendix. Document on release into turnover. Contract production in the Russian Federation

11.1 JSON format

Example:

```
{
  "producer_inn": "0000000000",
  "owner_inn": "2222222222",
  "participant_type": "owner",
  "production_date": "2019-04-10T00:00:00.000Z",
  "production_order": "CONTRACT_PRODUCTION",
  "products_list": [
    {
      "uit": "00000000000000000000000000000000",
      "uitu_code": "00000000000000000000",

```

```

    "tnved_code": "123456789",
    "production_date": "2019-04-10T00:00:00.000Z",
    "certificate_doc_type": "CONFORMITY_CERTIFICATE",
    "certificate_doc_number": "12345",
    "certificate_doc_date": "2019-04-10T00:00:00.000Z"
  }
]
}

```

Attribute content:

Name	Description	Type	Mandatory
producer_inn	INN of goods producer or importer	string	yes
owner_inn	INN of goods owner	string	yes
participant_type	type of participant involved into turnover admissible values for producer or owner	string	yes
production_date	Production date	string	yes
production_order	Production order type	string	yes
uit	Goods unique identifier	string	Mandatory if uitu is not specified
uitu	Shipment packing unique identifier	string	Mandatory if uit is not specified
tnved_code	10-	string	yes

	character code of EEU TNVED		
production_date	Goods production date (differs from the specified in the general data on goods release into turnover)	string	no
certificate_doc_type	Types mandatory certification documents	string	yes
certificate_doc_number	Conformity certificate number	string	yes
certificate_doc_date	Conformity certificate date	string	yes

12. Appendix. Document on release into turnover. Receival from physical entities.

12.1 JSON format

Example:

```
{
  "participant_inn": "1111111111",
  "product_receiving_date": "2019-12-04T00:00:00.000Z",
  "products_list": [
    {
      "uit": "00000000000000000000000000000000",
      "uitu_code": "00000000000000000000",
      "product_receiving_date": "2019-12-04T00:00:00.000Z"
    }
  ]
}
```

}

Attribute content:

Name	Description	Type	Mandatory
participant_inn	INN of turnover participant	string	yes
product_receiving_date	Goods receiving date	string	yes
uit	Goods unique identifier	string	Mandatory if uitu is not specified
uitu	Shipment packing unique identifier	string	Mandatory if uit is not specified

13. Appendix. Document on release into turnover. Cross-border trade

13.1 JSON format

Example:

```
{
  "trade_participant_inn": "00000000000",
  "sendex_tax_number": "123456",
  "exporter_name": "Company name",
  "country_oksm": "112",
  "import_date": "2019-04-10T00:00:00.000Z",
  "primary_document_number": "123",
  "primary_document_date": "2019-04-10T00:00:00.000Z",
  "products_list": [
    {
      "ki": "0000000000000000000000000000000000000000",
      "kitu": "",
      "tnved_code": "123456789",
      "cost": "100",
      "vat_value": "13",
      "certificate_doc_type": "CONFORMITY_CERT",
      "certificate_doc_number": "12345",

```

```

    "certificate_doc_date": "2019-04-10T00:00:00.000Z"
  }
]
}

```

Attribute content:

Name	Description	Type	Mandatory
trade_participant_inn	INN of turnover participant	string	yes
sender_tax_number	Consignor's taxpayer number	string	yes
exporter_name	Exporter name	string	yes
country_oksm	Country code as per OKSM	string	yes
import_date	Import date	string	yes
primary_document_number	Number of a primary document confirming goods movement	string	yes
primary_document_date	Date of a primary document confirming goods movement	string	yes
products_list	Goods list	string	yes
ki	Goods unique identifier	string	Mandatory if kitu is not specified
kitu	Shipment packing	string	Mandatory if ki is not

	unique identifier		specified
cost	Unit price	Floating point number	yes
vat_value	VAT value	Floating point number	yes
tnved_code	10-character code of EEU TNVED	string	yes
certificate_doc_type	Types of mandatory certification documents	string	yes
certificate_doc_number	Conformity certificate number	string	yes
certificate_doc_date	Conformity certificate date	string	yes

14. Appendix. Document on release into turnover. Marking of remaining amount

14.1 JSON format

```
{
  "trade_participant_inn": "0000000000",
  "products_list": [
    {
      "ki": "00000000000000000000000000000000",
      "kitu": "",
      "country": "000",
      "declaration_number": "number",
      "declaration_date": "2019-12-02T00:00:00.000Z",
      "certificate_type": "CONFORMITY_CERTIFICATE",

```

```

    "certificate_number": "number",
    "certificate_date": "2019-12-04T00:00:00.000Z"
  }
]
}

```

Attribute content:

Name	Description	Type	Mandatory
trade_participant_inn	INN of a turnover participant	string	yes
products_list	Goods parameters	array	yes
ki	Goods unique identifier	string	Mandatory if kitu is not specified
kitu	Shipment packing unique identifier	string	Mandatory if ki is not specified
declaration_number	Goods declaration number	string	no
declaration_date	Goods declaration date	string	no
country	Country of origin	string	no
certificate_type	Types of mandatory certification documents	string	no
certificate_number	Conformity certificate number	string	no
certificate_date	Conformity certificate date	string	no

15. Appendix. Document on goods shipment

15.1 JSON format

Example:

```

{
  "document_num": "0000000000",

```

```

"document_date": "2018-10-13T00:00:00.000Z",
"receiver_inn": "0000000000",
"receiver": "ИПНТ",
"sender_inn": "0000000000",
"sender": "string",
"owner_inn": "0000000000",
"owner": "ИПНТ",
"request_type": "SHIPMENT",
"turnover_type": "SELLING",
"transfer_date": "2018-10-13T00:00:00.000Z",
"pdf": "string",
"withdrawal_type": "string"
"withdrawal_date": "string"
"st_contract_id": "string",
"to_not_participant": true/false,
"products": [
  {
    "uitu_code": "00000000000000000000",
    "uit_code": "0000000000000000000000000000000000",
    "product_description": "string",
    "product_cost": 1000,
    "product_tax": 10
  }
]
}

```

After successful validation of a shipping document its status is applied for WAIT_ACCEPTANCE [Waiting for acceptance]. Once accepted the document status changes to CHECKED_OK.

Attribute content:

Name	Description	Type	Mandatory
document_num	primary document number	string	yes
document_date	primary document date	string	yes
transfer_date	date of marked goods hand over	string	yes
receiver_inn	INN of a consignee	string	yes
receiver	Consignee name	string	yes
owner_inn	INN of the owner	string	yes
owner	owner name	string	yes
turnover_type	shipment type. Possible values of SELLING- sale	string	yes
withdrawal_type	Reason for withdrawal from	string	no

	turnover		
withdrawal_date	Date of withdrawal from turnover	string	no
st_contract_id	State contract identifier	string	no
to_not_participant	attribute of shipment for a non-participant	string	yes
uit_code	Goods unique identifier	string	Mandatory if uitu is not specified
uitu_code	Shipment packing unique identifier	string	Mandatory if uit is not specified
product_description	product name	string	yes
product_cost	unit price	Floating point number	no
product_tax	VAT value	Floating point number	no

16. Appendix. Document on goods shipment cancellation

16.1 JSON format

Example:

```
{
  "participant_inn": "0000000000",
  "shipment_number": "string"
}
```

Attribute content:

Name	Description	Type	Mandatory
participant_inn	INN of a turnover participant	string	yes

shipment_number	number of a cancelled shipment document	string	yes
-----------------	---	--------	-----

17. Appendix. Document on goods acceptance

17.1 JSON format

Example:

```
{
  "request_type": "ACCEPTANCE",
  "release_order_number": "string",
  "products": [
    {
      "uit_code": "00000000000000000000000000000000",
      "product_description": "string",
      "product_cost": 100,
      "product_tax": 10
    }
  ],
  "document_date": "2018-12-31T21:00:00.000Z",
  "transfer_date": "2019-01-01T21:00:00.000Z",
  "document_number": "string",
  "trade_sender_inn": "0000000000",
  "trade_owner_inn": "0000000000",
  "trade_sender_name": "string",
  "trade_owner_name": "string",
  "trade_recipient_inn": "0000000000",
  "turnover_type": "SELLING",
  "acceptance_date": "2019-01-01T21:00:00.000Z",
  "release_method": "CROSSBORDER"
}
```

Attribute content:

Name	Description	Type	Mandatory
release_order_number	number of shipment request	string	yes
document_date	primary document date.	string	yes
transfer_date	goods shipment date	string	yes
document_number	primary document number	string	yes

trade_sender_inn	INN of a consignor	string	yes
trade_owner_inn	INN of a consignee	string	yes
trade_sender_name	name of a consignor organization	string	yes
trade_owner_name	name of a consignee organization	string	yes
turnover_type	shipment type. Possible values of SELLING- sale	string	yes, if release_method is not specified
acceptance_date	goods acceptance date	string	yes
"release_method"	release into turnover method Only for acceptance of a cross-border shipping document	string	no
uit_code	Goods unique identifier	string	Mandatory if uitu is not specified
uitu_code	Shipment packing unique identifier	string	Mandatory if uit is not specified
product_description	product name	string	yes
product_cost	unit price	Floating point number	no
product_tax	VAT value	Floating point number	no

18. Appendix. Document on goods withdrawal from turnover at sale to the final consumer

18.1 JSON format

Example:

```

{
  "order_number": "string",
  "order_date": "2018-09-17T21:00:00.000Z",
  "inn": "0000000000",
  "action": 1,
  "action_date": "2018-09-17T21:00:00.000Z",
  "document_type": 1,
  "document_number": "string",
  "document_date": "2018-09-17T21:00:00.000Z",
  "products": [
    {
      "uit_code": "00000000000000000000000000000000",
      "uitu_code": "00000000000000000000",
      "product_cost": 1000
    }
  ]
}

```

Attribute content:

Name	Description	Type	Mandatory
order_number	Request number on withdrawal from turnover.	string	yes
order_date	Date of request on withdrawal from turnover	string	yes
inn	INN of a turnover participant	string	yes
action	Type of withdrawal from turnover	string	yes
action_date	Date of withdrawal from turnover	string	yes
document_type	Type of document	string	yes
document_number	Document number	string	yes
document_date	Document date	string	yes
uit_code	Goods unique identifier	string	Mandatory if uitu is not specified
uitu_code	Shipment packing unique identifier	string	Mandatory if uit is not specified
primary_document_type	primary document type	string	yes
primary_document_number	primary document number	string	yes

primary_document_date	primary document date	string	yes
product_cost	goods price	string	yes

19. Appendix. Document on MC drop out with/without IM

19.1 JSON format

Example:

```
{
  "trade_participant_inn": "1111111111",
  "cancellation_reason": "KM_SPOILED",
  "cancellation_doc_date": "2019-02-09T21:00:00.000Z",
  "cancellation_doc_number": "document number",
  "km_list": [
    {
      "uit": "00000000000000000000000000000000",
      "uitu": "00000000000000000000"
      "cancellation_reason": "KM_SPOILED"
    }
  ]
}
```

Attribute content:

Name	Description	Type	Mandatory
trade_participant_inn	INN of a turnover participant	string	yes
cancelation_reason	Drop out reason	string	yes
cancellation_doc_date	document date confirming the drop out	string	yes
cancellation_doc_number	Document number confirming the drop out	string	yes
uit	Goods unique identifier	string	Mandatory if uitu is not specified
uitu	Shipment packing unique identifier	string	Mandatory if uit is not specified

cancellation_reason	Drop out reason	string	yes
---------------------	-----------------	--------	-----

20. Appendix. Document on goods remarking

20.1 JSON format

Example:

```
{
  "participant_inn": "00000000000",
  "remarking_date": "2018-09-17T21:00:00.000Z",
  "remarking_cause": "DAMAGED_OR_LOST",
  "products": [
    {
      "certificate_document": "CONFORMITY_CERT",
      "certificate_document_number": "certificateDocumentNumber",
      "certificate_document_date": "2018-09-17T21:00:00.000Z",
      "last_uin": "lastUin",
      "new_uin": "newUin",
      "remarking_date": "2018-09-17T21:00:00.000Z",
      "remarking_cause": "DAMAGED_OR_LOST"
    }
  ]
}
```

Attribute content:

Name	Description	Type	Mandatory
participant_inn	INN of a turnover participant	string	yes
remarking_date	Remarking date	string	yes
remarking_cause	Remarking reason DAMAGED_OR_LOST (IM with MC is damaged or lost), ERROR_DETECTED (Goods description errors are revealed)		
certificate_document	Type of mandatory certification document. CONFORMITY_CERT - conformity certificate, CONFORMITY_DECLAR - conformity declaration	string	yes

certificate_document_number	Number of mandatory certification document	string	yes
certificate_document_date	Date of mandatory certification document	string	yes
last_uin	Previous goods unique identifier	string	yes
new_uin	New goods unique identifier	string	yes
remarking_date	Remarking date	string	no
remarking_cause	Remarking reason DAMAGED_OR_LOST (IM with MC is damaged or lost), ERROR_DETECTED (Goods description errors are revealed)	string	no

21. Appendix. Document on shipping package aggregation

21.1 JSON format

```
{
  "participantId":"00000000000",
  "aggregationUnits":[
    {
      "unitSerialNumber":"200001111100001111",
      "aggregationType":"AGGREGATION",
      "sntins":[
        "00000000000000000000000000000000", "000001111100001112"
      ]
    }
  ]
}
```

Attribute content:

Name	Description	Type	Mandatory
participantId	INN of a turnover participant	string	yes
aggregationUnits	Array of created aggregates	array	yes
unitSerialNumber	Aggregate SSCC code	string	yes

aggregationType	aggregation type shall be "AGGREGATION"	string	yes
sntins	MC in an aggregate	array	yes

22. Appendix. Document on shipping package disaggregation

22.1 JSON format

```
{
  "participant_inn": "5834052090",
  "products_list": [
    {
      "uitu": "147600877000000010"
    }, {
      "uitu": "147600877000000055"
    }, {
      "uitu": "147600877000000132"
    }
  ]
}
```

Attribute content:

Name	Description	Type	Mandatory
participant_inn	INN of a turnover participant	string	yes
products_list	Array of aggregates for disaggregation	array	yes
uitu	Aggregate SSCC code	string	yes

23. Appendix. Document on shipping package re-aggregation

23.1 JSON format

```
{
  "participant_inn": "0000000000",
  "reaggregation_type": "REMOVING",
  "uitu": "1111111111111111",
  "uit_uitu_list": [
    {
      "uit_uitu": "00000000000000000000000000000000"
    }, {
      "uit_uitu": "00000000000000000000000000000000"
    }, {
      "uit_uitu": "11111111111111111111111111111111"
    }
  ]
}
```

Attribute content:

Name	Description	Type	Mandatory
participant_inn	INN of turnover participant	string	yes
reaggregation_type	Operation type: REMOVING - withdrawal, ADDING - adding	string	yes
uitu	Modified aggregate code	string	yes
uit_uitu_list	Array of aggregates for adding/withdrawal	array	yes
uit_uitu	Marking codes for adding/withdrawal	string	yes

24. Appendix. Document on remaining goods description

24.1 JSON format

```
{
  "trade_participant_inn": "0000000000",
  "products_list": [
    {
      "tnved_code_2" : "62",
      "brand": "Торговая марка",

```

```

        "name": "Наименование товара",
        "product_gender": "MALE",
        "release_method": "PRODUCED_IN_RF",
        "consumer_age": "Возрастная категория",
        "model": "Модель"
    }
}
}

```

Attribute content:

Name		Description	Type
trade_participant_inn		INN of a turnover participant	string
products_list		Goods parameters	array
	tnved_code_2	Goods nomenclature code (2 characters)	string
	brand	Trade brand	string
	name	Goods name	string
	product_gender	Footwear, gender category	guid
	release_method	Goods turnover release method	guid
	consumer_age	Age category	string
	model	Model	string

25. Examples:

25.1 Authentication by EQES/Documents signing

25.1.1 Example of a security token obtaining at login with attached signature/signing of documents with detached signature in 1C

```

// sThumbprint - отпечаток сертификата, используемого для подписи; строка,
// представляющая отпечаток в шестнадцатеричном виде
// пример 195934d72dcdf69149901d6632aca4562d8806d8
// ТекстДляПодписи должен быть в Base64
// bDetached - Истина/Ложь - откреплённая (для подписания
документов) /прикреплённая (для получения токена авторизации) подпись

```

```
Функция ПодписатьТекст(ТекстДляПодписи, sThumbprint, bDetached)
CADESCOM_BASE64_TO_BINARY = 1; // Входные данные пришли в Base64
CADESCOM_CADES_TYPE = 1; // Тип усовершенствованной подписи
CAPICOM_AUTHENTICATED_ATTRIBUTE_SIGNING_TIME = 0; // Атрибут штампа времени
подписи
```

```
oSigner = Новый СОМОбъект("CADESCOM.CPSigner");
// Объект, задающий параметры создания и содержащий информацию об
усовершенствованной подписи.
oSigner.Certificate = ПолучитьСертификатПоОтпечатку(sThumbprint);

oSigningTimeAttr = Новый СОМОбъект("CADESCOM.CPAttribute");
oSigningTimeAttr.Name = CAPICOM_AUTHENTICATED_ATTRIBUTE_SIGNING_TIME;
oSigningTimeAttr.Value = ТекущаяДата();
oSigner.AuthenticatedAttributes2.Add(oSigningTimeAttr);
```

```
ТекстДляПодписи = СокрЛП(ТекстДляПодписи);
```

```
oSignedData = Новый СОМОбъект("CADESCOM.CadesSignedData");
// Объект CadesSignedData предоставляет свойства и методы для работы с
усовершенствованной подписью.
oSignedData.ContentEncoding = CADESCOM_BASE64_TO_BINARY;
SignedData.Content = СокрЛП(ТекстДляПодписи);
EncodingType = 0;
```

```
sSignedMessage = oSignedData.SignCades(oSigner, CADESCOM_CADES_TYPE,
bDetached, EncodingType);
// Метод добавляет к сообщению усовершенствованную подпись.
Возврат sSignedMessage; // Подпись в формате Base64
КонецФункции
```

```
//Отпечаток - строка HEX
```

```
Функция ПолучитьСертификатПоОтпечатку(ОтпечатокСтр)
Рез = Неопределено; // Найденный сертификат (Сом-объект)
CAPICOM_CURRENT_USER_STORE = 2;
//2 - Искать сертификат в ветке "Личное" хранилища.
CAPICOM_MY_STORE = "My";
```

```
// Указываем, что ветку "Личное" берем из хранилища текущего пользователя
CAPICOM_STORE_OPEN_READ_ONLY = 0; // Открыть хранилище только на чтение
oStore = Новый СОМОбъект("CADESCOM.Store"); // Объект описывает хранилище
сертификатов
oStore.Open(CAPICOM_CURRENT_USER_STORE, CAPICOM_MY_STORE,
CAPICOM_STORE_OPEN_READ_ONLY); // Открыть хранилище сертификатов
```

```
// 1 вариант: поиск сертификата по отпечатку
CAPICOM_CERTIFICATE_FIND_SHA1_HASH = 0;
Certificates = oStore.Certificates.Find(CAPICOM_CERTIFICATE_FIND_SHA1_HASH,
ОтпечатокСтр);
Рез = Certificates.Item(1);
```

```
//2 вариант: обходом по коллекции и сравнение с отпечатком
//Для Каждого ТекСертификат Из oStore.Certificates Цикл
//ТекОтпечаток = ТекСертификат.Thumbprint; // возвращается отпечаток в
шестнадцатеричном виде
//Если ВРЕГ(ТекОтпечаток) = ВРЕГ(ОтпечатокСтр) Тогда //Рез = ТекСертификат;
```

```

//Прервать;
//КонецЕсли;
//КонецЦикла;
oStore.Close(); // Закреть хранилище сертификатов и освободить объект 61

```

```

Возврат Рез;
КонецФункции

```

25.1.2 Example of a security token obtaining at REACT JS login.

```

import { CadesPluginer as anyName } from "@crpt/cades-pluginer"; //or
import CadesPluginer from "@crpt/cades-pluginer";
//usually you will need 2 methods
CadesPluginer.getFinalCertsArray().then(certs => {
  console.log("certs", certs);
  CadesPluginer.signMessage("lalala", certs[0].certificate).then(signed =>
    console.log("signed", signed);
  );
});

```

25.1.3 Example of a security token obtaining at login with attached signature/signing of documents with a detached signature on Java.

```

String data = ...//получение кода для авторизации
boolean detached = false;
// false - прикреплённая(для получения токена авторизации) подпись
// true - откреплённая(для подписания документов) подпись

//Формирование подписи
KeyStore keyStore = loadKeyStore();//инициализация хранилища ключей
List<X509CertificateHolder> chain = new ArrayList<>();
List<Certificate> certs = Arrays.asList(keyStore.getCertificateChain(alias));
certs.forEach(cert -> {
  try {
    chain.add(new X509CertificateHolder(cert.getEncoded()));
  } catch (IOException | CertificateEncodingException e) {
    log.error("Error while building certificate chain", e);
  }
});
PrivateKey privateKey = (PrivateKey) (keyStore.getKey(alias, password));
ByteArrayOutputStream out = new ByteArrayOutputStream();
CAdESSignature signature = new CAdESSignature(detached);
signature.setCertificateStore(new CollectionStore(chain));

final Hashtable table = new Hashtable();
Attribute attr = new Attribute(CMSAttributes.signingTime,
  new DERSet(new Time(new Date()))); // устанавливаем время
подписи

table.put(attr.getAttrType(), attr);

AttributeTable attrTable = new AttributeTable(table);

```

```

//Добавление подписанта. Алгоритмы могут отличаться в зависимости от
требований
signature.addSigner(JCP.PROVIDER_NAME,
                    JCP.GOST_DIGEST_OID,
                    JCP.GOST_EL_DH_OID,
                    privateKey,
                    certs,
                    CAdESType.CAdES_BES,
                    null,
                    false,
                    attrTable,
                    null);

signature.open(out);
signature.update(data);
signature.close();
byte[] signedCode = out.toByteArray();
//Использовать подписанный код для получение токена / сформированную
открепленную подпись УКЭП документа вместе с самим документом

```

Description of plugin methods:

Method	Description
getFinalCertsArray(): Promise<Cert[]>	Init plugin, create store, . Cert: {certificate: {}}, info: [] }, certificate is an original cadesplugin Certificate object, info is an array [name, date_from, date_to, serial_number].
signMessage(message, cert, detached): Promise	Sign message with selected certificate.