Cloud caching device protocol

(OrangeData fiscal cloud)

Version 2.28.0 dated 16.08.2021

##### Contents

##### [1. General description](#_1._General_description)

##### [1.1. Basic information](#_1.1._Basic_information)

##### [1.2. Connection to fiscal cloud](#_1.2._Connection_to)

##### [1.2.1. Two test environments](#_Toc59045179)

##### [1.2.2. Production environment](#_Toc59045180)

##### [1.2.3. Security](#_Toc59045181)

##### [1.2.4. SSL certificates](#_Toc59045182)

##### 1.2.5. Request body signing

##### [1.2.6. Other connection options](#_Toc59045184)

##### [1.3. Request with Signature Example](#_Toc59045185)

##### [1.4. Testing API with Postman and Signature Generator application.](#_Toc59045186)

##### [2. Requests](#_Toc59045187)

##### [2.1 Receipt creation](#_Toc59045188)

##### [2.1.1 Request Body](#_Toc59045189)

##### [2.1.1.1 Document contents](#_Toc59045190)

##### [2.1.1.2 Subject of Settlement](#_Toc59045191)

##### [2.1.1.3 Receipt Close Parameters](#_Toc59045192)

##### [2.1.1.4 Payment](#_Toc59045193)

##### [2.1.1.5 Additional user details](#_Toc59045194)

##### [2.1.1.6 Supplier Data](#_Toc59045195)

##### [2.1.1.7 Agent Data](#_Toc59045196) [2.1.1.8 Customer (customer) information, 1256](#_2.1.1.8_Customer_(customer))

##### [2.1.1.9 Operational details of the check, 1270](#_2.1.1.9_Operational_details) [2.1.1.10 Industry check requisite, 1261](#_2.1.1.10_Industry_check)

##### [2.1.1.11 Product code, 1163](#_2.1.1.11_Product_code,)

##### [2.1.1.12 Fractional quantity of labeled goods, 1291](#_2.1.1.12_Fractional_quantity)

##### [2.1.2 Response body](#_2.1.2_Response_body)

##### [2.2 Receipt status](#_2.2_Receipt_status)

##### [2.2.1 Response body](#_Toc59045199) [2.2.2 Response body with CM processing errors](#_2.2.2_Response_body)

##### [2.2.2.1 Error processing the subject of calculation containing CM](#_2.2.2.1_Error_processing)

##### [2.2.2.2 Result of KM check](#_2.2.2.2_Result_of)

##### [2.2.2.3 Server response to the CM check request](#_2.2.2.3_Server_response)

##### [2.3 Creating a correction receipt](#_Toc59045200)

##### [2.3.1 Request body](#_Toc59045201)

##### [2.3.1.1 Document content](#_Toc59045202)

##### [2.3.2 Response body with request processing errors](#_Toc59045203)

##### [2.4 Correction Receipt Status](#_Toc59045204)

##### [2.4.1 Response Body](#_Toc59045205)

##### [2.5 Cashing machine status in the group](#_Toc59045206)

##### [2.5.1 Response Body](#_Toc59045207)

##### [2.5.1.1 The structure of the object information about the device](#_Toc59045208)

##### [2.5.2 Request processing errors response body](#_Toc59045209) [2.6 Checking access parameters](#_2.6_Checking_access)

##### [2.6.1 Request body](#_2.6.1_Request_body)

##### [2.6.2 Response Body with Request Processing Errors](#_2.6.2_Response_body)

##### [2.7 Creating a check for FFD correction 1.2](#_2.7_Creating_a)

##### [2.7.1 Request Body](#_2.7.1_Request_body)

##### [2.7.1.1 Document content](#_2.7.1.1_Document_content)

##### [2.7.2 Response body with request processing errors](#_2.7.2_Response_body)

##### [2.8 Status of FFD correction receipt 1.2](#_2.8_Status_of)

##### [2.8.1 Response body](#_2.8.1_Response_body)

##### [2.9 Checking the marking code](#_2.9_Checking_the)

##### [2.9.1 Request Body](#_2.9.1_Request_body)

##### [2.9.1.1 Document Content](#_2.9.1.1_Document_content)

##### [2.9.2 Response body with request processing errors](#_2.9.2_Response_body)

##### [2.10 Check status of marking code](#_2.10_Marking_code)

##### [2.10.1 Response body](#_2.10.1_Response_body) [2.11 Status of FFD correction receipt 1.2](#_2.11_Status_of)

##### [2.11.1 Response body](#_2.11.1_Response_body)

##### [3. Swagger](#_Toc59045210)

##### [4. Links to the receipt.](#_Toc59045211)

##### 5.Change Log

# 1. General description

## 1.1. Basic information

**Orange data fiscal cloud** – is a service that allows users generate fiscal receipts for internet commerce. A client can register in [https://orangedata.ru/](https://orangedata.ru/%20%20)  website, order several physical fiscal devices connected to his account and generate fiscal receipts, using the API (described below). The service generates receipts in electronic form only (without printed hard copy).

**Caching device** – special device that meets requirements of of Russian Federal law of 22.05.2003 N 54-FZ (edited on 03.07.2018) «Сash equipment usage and settlements in the Russian Federation». According to the law, all caching devices are listed in special roster of cash registers. There is a range of different caching devices (with human interface, with hardware RS-232 or USB interface, e.t.c). A caching device can generate receipts in paper-printed form or in electronic form. The main COMMON features of all caching devices are:

* Caching device must correctly communicate with special fiscal storage (see description below).
* Each receipt must contain mandatory fields: fiscal storage serial number (**fsNumber**), device serial number (**deviceSN**), device register number (**deviceRN**), INN, shift number, document index, fiscal sign. All these fields are described below.

Orange fiscal cloud is based on cashing devices “NETPAY-ФС” (details are here: <http://paykiosk.ru/netpay> ). In each 2 units in server rack there are 196 caching devices are installed and clients can “reserve” one or more caching devices for their business. Each caching device can be registered in tax authority ONLY for one company. So, caching devices can be shared between several departments of one company, but not between several clients.

|  |  |
| --- | --- |
| http://paykiosk.ru/images/slider/netpay-1-1.png | **Picture 1.**  NETPAY-ФС *caching device with fiscal storage inserted.*  *Devices can be mounted into a rack.* |

**Device serial number** (field **deviceSN**) - is serial number of caching register device, assigned by manufacturer. According to the law, it  consists of 16 digits. Each caching register is registered in tax authority.

**Device register number** (field **deviceRN**) - is the number that tax authority assigned to the caching register when  the company register the device in the tax authority. It contains 16 digits. Device register number and device serial number – are different. **deviceSN** never changes. **deviceRN** can changed when a company unegister caching device in Tax Authority and (after a pause) register the caching device again.

**Fiscal storage** (abbreviation: FN – “*fiscalny nakopitel*”)– is special storage device with Java machine inside and hardware security.  Fiscal storage is integrated into any caching register and controls key functionality of cashing register. Fiscal storage allows WRITE-AND-READ-WITHOUT-MODIFICATION operations. So nobody can change a committed transaction. Any correction - is stored as a new transaction in fiscal storage.

Each fiscal storage has unique number, containing 16 digits.

|  |  |
| --- | --- |
|  | **Picture****2***.*  *Production fiscal storage. The size of device is 30 mm.* |

**Fiscal storage number (fsNumber**) – unique 16-digit fiscal storage serial number, assigned by manufacturer.

Fiscal storage numbers starting with four digits “9” (for instance 9999078900006784) are reserved for test fiscal storage devices. All other prefixes are used for production environment.

**INN** (abbreviation 'identificator nalogoplatelshika' = tax payer identifier) an unique identifier that tax authority assign to each tax payer. It contains 10 digits for companies and 12 digits for physical persons and entrepreneurs. For instance, INN 7725327863 is identifier of “Orange Data” company.

**shiftNumber** - the number of shift. When a cashier start working, it opens shift. Fiscal storage generates a special transaction and report this fact to tax authority. At the end of the working day the cachier closes his shith and this reported to tax authority too. A new fiscal storage starts with shiftNumber=1. The next is 2 and so on.

**documentIndex**- the number of document inside a SHIFT. When a cashier opens a shift, it generates a documentIndex=1 and next document is 2 and so on...

**documentNumber**- unique document number that FN assigns to any document. A new FN generates the first document with documentNumber=1, the next document is 2 and so on…

**Fiscal sign** (field **fp** - in Russian – “fiscalny priznak”)- is a "check sum" that is generated by Fiscal storage based on: date, amount, INN field, fiscal storage number, device register number, fiscal number. Tax authority can use **fp** field to verify whether the receipt was legal or false. If it identify falsification - it does not accept receipts from caching register and the caching device will be locked in 30 days.

**FFD** (format of fiscal data) – format of binary document that caching device generates and sends to Tax Authority. Actual FFD are: **FFD 1.05** and **FFD 1.2.**

**OFD** (Operator of Fiscal Data) – is organization responsible for receiving fiscal documents in binary format and sending information to Tax Authority. There are about 15 authorized ODF in Russia.

## 1.2. Connection to fiscal cloud

OrangeData has two test environments and one production environment.

### 1.2.1. Two test environments

**Differences between test environment 1 and test environment 2 are described below.**

|  |  |
| --- | --- |
| **Test environment 1.**  This test environment provides API functionality without web cabinet. Device emulators are used instead of physical devices. Any tester can use OrangeData test environment 1 for free.  Test environment 1 can be used for any client’s INN without long configuration period.  Test environment 1 is free of charge and recommended for clients’ tests. | **Test environment 2.**  This test environment provides both web cabinet and API functionality. Test environment 2 uses physical devices equipped with test fiscal storage (MGM).  Test environment 2 setup requires 1-3 days for configuration and device installation for each company. It is not a free service.  Users cannot use any INN for tests. Only specific INN can be configured for each test physical device.  Test environment mostly used by OrangeData developers and not recommended for clients (some functionality can be down when OrangeData team is testing new releases). |

**Connection settings test environments:**

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Test environment 1** | **Test environment 2** |
| Base URL | <https://apip.orangedata.ru:2443> | <https://apip.orangedata.ru:12001> |
| IP address | 94.228.252.55 | 94.228.252.55 |
| Web Cabinet | Not available | https://test.orangedata.ru/lk/ |
| Availability check URL for browsers | https://apip.orangedata.ru:2443/api/v2/ | https://apip.orangedata.ru:12001/api/v2/ |
| Swagger-ui | <https://apip.orangedata.ru:2443/swagger> | <https://apip.orangedata.ru:12001/swagger> |
| Swagger JSON specification | <https://apip.orangedata.ru:2443/swagger/v2/swagger.json> | <https://apip.orangedata.ru:12001/swagger/v2/swagger.json> |

### 1.2.2. Production environment

|  |  |
| --- | --- |
| **Parameter** | **Production environment** |
| Base URL | <https://api.orangedata.ru:12003> |
| IP address | 62.76.112.48 and 188.170.11.161 |
| Web Cabinet | <https://lk.orangedata.ru> |
| Availability check URL for browsers | It is highly recommended to use domain name for production requests.  Add /api/v2/ to the URL bar to check availability |
| Swagger-ui | <https://api.orangedata.ru:12003/swagger> |
| Swagger JSON specification | <https://apip.orangedata.ru:12003/swagger/v2/swagger.json> |

### 1.2.3. Security

Security is provided by two steps:

1. Each client generates his own SSL certificate. This SSL certificate provides secure connections between client’s application and OrangeData fiscal cloud. In production environment SSL certificate is generated in web cabinet <https://lk.orangedata.ru/> In two test environments the default test certificate is used. You can download test certificate from Git here <https://github.com/orangedata-official/API/tree/master/files_for_test/client_ca.crt>
2. Any request to OrangeData fiscal cloud that POST any data mast be signed. For instance, document creation – need signature. At the same time – request “**get receipt status**” (GET request) does not need signature. More details about document signing you can read bellow in sections: **1.2.5** and **1.3**.

### 1.2.4. SSL certificate and keys for test environment

In both test environments (1 and 2) https connection with certificates is used. All certificates are published in Git here <https://github.com/orangedata-official/API/tree/master/files_for_test>

**File description in GIT file\_for\_test directory**

|  |  |
| --- | --- |
| **File** | **Description** |
| **client\_ca.crt** | Root OrangeData certificate (with public key inside). All client’s certificates are signed by corresponding private key, related to this public certificate. If a client’s certificate was signed by any other issuer – it will be rejected by server |
| **client.key** | Client’s private key for SSL connection for test environment |
| **client.crt** | Client’s public certificate (issued by root OrangeData certificate – see client\_ca.crt) for test environment |
| **сlient.pfx** | Client’s PFX (PKCS # 12) certificate for test environment |
| **private\_key.xml** | Test client’s private key for signing request body. More details are described below in section *[1.2.1.4. Request body signing]* |
| **readme\_rus.txt** | Readme in Russian language |
| **readme\_eng.txt** | Readme in English language |

**File description in GIT SignatureGenerator directory**

|  |  |
| --- | --- |
| **File** | **Description** |
| SignatureGenerator.exe.config | Signature generator config file |
| SignatureGenerator.exe | Signature generator executable |
| SignatureGenerator.pdb | Signature generator data file |

### 1.2.5. Request body signing

Some requests (for instance – all fiscal document creation requests) require body signing. The signature mast be added to HEADER parameter **X-Signature**.

The SHA256-RSA signature with a key of at least 2048 Kbps and Pkcs1 padding is used to protect client messages from changes. The signature is transmitted in the X-Signature http-header encoded using base64 encoding.

For both test environments file **private\_key.xml** from Git here <https://github.com/orangedata-official/API/tree/master/files_for_test> mast be used for signing.

### 1.2.6. Other connection options

Interaction with cloud caching device is performed by web HTTPS service. HTTPS server uses client and server certificates for mutual authentication.

Message format is JSON. JSON size has no limitation, but any Fiscal Storage has hardware limitation 32768 bytes for each fiscal document in binary form. This document size limitation provides approximately 250 positions in a receipt. Fiscal Storage will reject your fiscal document if you exceed binary limitation 32768 bytes. We recommend you use less than 250 positions in a receipt.

Caching devices generate documents according to **FFD 1.05** and **FFD 1.2**. See more details on Tax Authority web site <https://www.nalog.ru/html/sites/www.new.nalog.ru/doc/pril2_fns229_210317.docx>

Fiscal cloud API uses UTF-8 encoding, but all types of **fiscal storages** support only **CP866** encoding and fiscal cloud converts receipt data from JSON with **UTF-8** encoding to Tag-Length-Value (TLV) structure with **CP866** encoding. So, a restriction on encoding of transmitted data appears: fiscal cloud transmits messages using **UTF‑8**, but FS uses **CP866** encoding with smaller set of characters than **UTF-8**. If the system receives an “unsupported” character in **CP866** – the character will disappear from final fiscal document (for instance there are no angle quotes ‘«’ and ‘»’ in **CP866**  encoding and FS will replace ‘«Dream» Juice’ to ‘Dream Juice’).

Cashing device does not perform full-fledged format-logical data control. For instance, according to FFD, the tag 1008 (Customer Phone Number or Customer Email) have format of +{N} or {X}@{X}, but the Tax Authority and FN and our system will accept any string with the length from 1 to 64 symbols. There is no use to perform more strict control than Tax Authority does. It is recommended to familiarize yourself with the FFD <https://www.nalog.ru/html/sites/www.new.nalog.ru/doc/pril2_fns229_210317.docx> Monitor formats and the presence/absence of tags, depending on your specific work scenario.

Please take care of correct processing of network connection errors. If you got an error when sending a fiscal document, please perform sending several times with some timeout. We recommend 5 attempts with an interval of 10-20 seconds (the figures are for illustration purposes and may vary depending on the specific situation).

## 1.3. Request with Signature Example

POST /api/v2/documents/ HTTP/1.0

Connection: close

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

Accept: \*/\*

Host: apip.orangedata.ru

Content-Length: 509

X-Signature: Ql1ZoZfYQo9NAgJQXUsBKQHQjqkICmn9b2jCfUyUDQYExMRtfhciGEs35dUFisVWiveeWF9dgBkzbJtNgEyQoNRXE+37ZMmEjS/SizRVpeAIKfpAx8GseQnv3ssw2NtvAELMduAgsz3hC8mIrOt//CPmQABd8X21NqfzZY4zVZgOTu/q2fNXBlv+UyvJ5amwzMkh7aX5g0RQDBAQ7UovscpQY41t43rdYZ6UVVv/OHEfExs7vEZmi7c3tdUJVf98Bli3oh5RfMnsb1/m2kZBqhxC0zThCKdyGlmFNM+tAlVqbKHDkulZRgwarwXbL17y/xbPeLlQZLzIU2esYTMulg==

{“id”:”2\_loc\_z5bYWHvD”,”inn”:”1234567890”,”group”:”Main”,”key”:”1234567890”,”content”:{“type”:1,”positions”:[{“quantity”:2.0,”price”:10.0,”tax”:1,”text”:”\u0421\u0430\u043c\u043e\u0432\u044b\u0432\u043e\u0437 ru”,”paymentMethodType”:4,”paymentSubjectType”:1},{“quantity”:1.0,”price”:0.0,”tax”:1,”text”:”\u0414\u043e\u0441\u0442\u0430\u0432\u043a\u0430”,”paymentMethodType”:4,”paymentSubjectType”:4}],”checkClose”:{“payments”:[{“type”:2,”amount”:20.0}],”taxationSystem”:0},”customerContact”:”user@domain.com”}}

RSA key:

<RSAKeyValue>

  <Modulus>t8nC/Eth8UabQbXu8pdro3v7NqUanV8Y+g92YgT7z1xqkBLRHXZ1guml3PxrqjNX9AvOmu8R+qaKOyHfJW0PcRDLzCoIUcHNAwpDO/E5j6WaaLIv7gAjTtyr9kJB9rfJaparViJNZu3RSUYGTvVznOmXMf7LTOTMR6HP/5H1TP5n1g4+BbLmC9EhjUf2eNFqwZBqPtzybBb6jaHBRaJ0XdE3lh2OeE9/OF0BtLwiYPDKsVTxIekbNf7l/DREy+YbUOxQLceeHXrvbYLiGWecP0a7CqHGj9ZNY1oJThK3AwrSd4yHa9Wnx/GaZUNtWud1BaP9g3sVX+sRV9xtnI96dw==</Modulus>

  <Exponent>AQAB</Exponent>

  <P>3WSb72a1erb6jcLkyZA2Y21VNIipGz+ta1RP+iacs3xnktFsxgTYgqWyt6SWZ2rStp0u4vb/IAHyKhgJPNTUSi2u0G44MosRxMC/FWTF8zdyrDF4BjPBM4j84nAmE/FQYv5F8ldDkakc96zEPiTk5Fka3MpeN8mMk6/OA59JdF0=</P>

  <Q>1IRVid5SsDrOwJQAEKkdT436Xeb0sVWe9AcU8JyaCEEMj0NpzownNbIrebPofMYdDHikopQpr2XqxZYDbb7AneoHkhEV26TfpPVbN4wBJFXih3lAP2n5hqhgqHGp5Wq2Lu7jUS376Ruw3bhwW+MiWpXv1xhMTZ8AtDfnZFFNvOM=</Q>

  <DP>Fo5KiNCJCtCbpFfH4XVM5UJdXPXTbNBHBdlYMJ9AddTl5Ijrt50ExgLFu4oMPMsYXryS61LI2WT5XCqIvmbcnhYbambgWLOKYuZUUYSr2kS67So5FUCunWaGhTdx2bRLQVqwm6kiXDPDnMRAViiCHXWqk/VsrXheVymhLqNK440=</DP>

  <DQ>mowSWMzhfV+G8+2tjnAt7KjnpSvEzyHhEr4DsGdybQZBR/4/j4nFCfukOkFnlTXN8j/aGpF9Lx0C+uX5YfoUYcLL9qGOL8lbCu+TgnXCbtY2gybeXj+HqzI3+MeQMlLEYqU/ks3KIOAOY2+55ljrpszbOqVk+B3luSnekMm/qtk=</DQ>

  <InverseQ>aP5e5F1j6s82Pm7dCpH3mRZWnfZIKqoNQIq2BO8vA9/WrdFI2C27uNhxCp2ZDMulRdBZcoeHcwJjnyDzg4I4gBZ2nSKkVdlN1REoTjLBBdlHi8XkiXzxvpItc2wjNC2AKHaJqj/dnh3bbTAQD1iUAxPmmLJYYkhfZ2i1IrTVxZE=</InverseQ>

  <D>PufM+Aq6kZSVWAetsL3EajKAxOuwQCDhVx+ovW4j+DQ8Y+WiTEyfShNV9qVD0PBltz3omch1GjpFhQn6OaRvraeIDH9HXttb3Fojr2zzYG4yrrYbPSRWoYj63ZwiIP2O7zdl0caGQHezfNcYa2N0NTG99DGc3/q6EnhlvjWQsSbiEjmxcPx8fmV1i4DoflMQ383nsixAFapgrROUAtCgMvhWn1kSeoojKd+e4eKZxa/SNYulsBJWNFkmo1CZH4YtqlPM+IwYeDUOnOUGNxGurRZ3qQdWs2N2ZQhnrvlh+zpzurD2hwAz6gQXP7mxxMR1xHtAD8XQ+w4OiJK6VwjoIQ==</D>

</RSAKeyValue>

Signature generation in php using <https://github.com/phpseclib/phpseclib>

function computeSignature($data)

{

    $rsa = new RSA();

    $rsa->setPrivateKey(($this->privateKey);

    $rsa->setPrivateKeyFormat(RSA::PRIVATE\_FORMAT\_XML);

    $rsa->setHash(‘sha256’);

    $rsa->setMGFHash(‘sha256’);

    $rsa->setSignatureMode(RSA::SIGNATURE\_PKCS1);

    return base64\_encode($rsa->sign($data));

}

Example of signature generation in php using openssl. PEM key is required for this method. You can convert xml into PEM using converter <https://superdry.apphb.com/tools/online-rsa-key-converter>:

function computeSignature($data)

{

    $key = <<<KEYDATA

    -----BEGIN RSA PRIVATE KEY-----

    MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQC3ycL8S2HxRptB

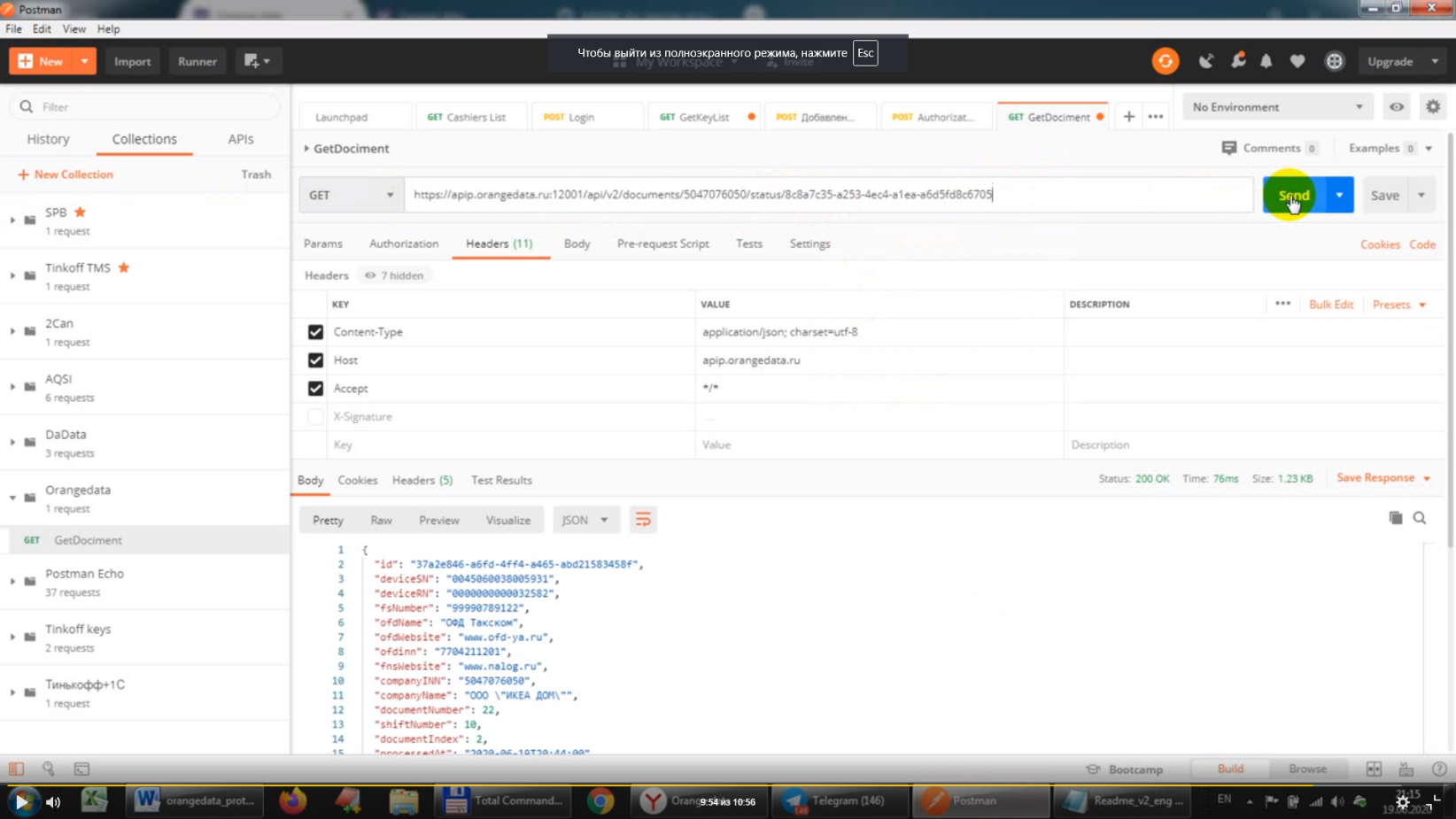
    te7yl2uje/s2pRqdXxj6D3ZiBPvPXGqQEtEddnWC6aXc/GuqM1f0C86a7xH6poo7

    Id8lbQ9xEMvMKghRwc0DCkM78TmPpYBosi/uACNO3Kv2QkH2t8lqlqtWIk1m7dFJ

    RgZO9Xoc6Zcx/stM5MxHoc//kfVM/mfWDj4FsuYL0SGNR/Z40WrBkGo+3PJsFvqN

    ocFFonRd0TeWHY54T384XQG0vCJg8MqxVPEh6Rs1/uX8NETL5htQ7Fatx54deu9t

    guIZZ5w/RrsKocaP1k1jWglOErcDCtJ3jIdr1afH8ZplQ21a53Ufo/2DexVf6xFX

    3G2cj3p3AgMBAAECggEAPUfM+Aq6kZSVWAetsL3EajKAxOuwQCDhVx+ovW4j+DQ8

    Y+WiTEyfShNV9qVD0PBltz3omch1GjpFhQn6OaRvraeIDH9HXttb3Fojr2zzYG4y

    rrYbPSRWoYj63ZwiIP2O7zdl0caGQHezfNcYa2N0NTG99DGc3/q6EnhlvjWQsSbi

    EjmxcPx8fmV1i4DoflMQ383nsixAFapgrROUAtCgMvhWn1kSeoojKd+e4eKZxa/S

    NYulsBJWNFkmo1CZH4YtqlPM+IwYeDUOnOUGNxGurRZ3qQdWs2N2ZQhnrvlh+zpz

    urD2hwAz6gQXP7mxxMR1xHtAD8XQ+w4OiJK6VWjoIQKBgQDdZJvvZrV6tvqNwuTJ

    kDZjbVU0iKkbP61rVE/6JpyzfGeS0WzGBNiCpbK3pJZnatK2nS7i9v8gAfIqGAk8

    1NRKLa7Qbjgw6xHEwL8VZMXzN3KsMXgGM8EziPzicCYT8Vbi/kXyV0OrqRz3rMQ+

    JOTkWRrcw943yYyTr84Dn0l0XQKBgQDUhFWJ3lKwOs7AlAAQqR1PjfpcRvSxVZ70

    BxTwnJoIQQyPQ0/OjCc1sit5s+h8xh0MeKSilCmvZerFlgNtvsCd6geSERXbpN+k

    9Vs3jAEkVeKHeUA/afmGqGCocanlarYu7uNRLfvpG7DduHBb4yJale/XGExNnwC0

    N+dkUU284wKBgBaOSojQiQrQm6RXx+F1TOVCXVz102zQRwXZWDCfQHXU5eSCa7ed

    BMYCxbuKDDzLGF68kutSyNlk+VwqiL5m3J4WG2pm4FizimLmVFGEq9pEuu0qORVA

    rp1mhoU3cdm0S0FasJupIlwzw5zEQFYogh11qpP1bK14XlcpoS6jSuONAoGBAJqM

    EljM4X1fhvPtrY5wLeyo56UrxM8h4RK+A7Bncm0GQUf+P4+JxQn7pDpBZ5U1zfI/

    2hqRfS8dAvrl+WBaFGHCy/ahji/JWwrvk4J1wm7WnoMm3l4/h0MyN/jHkDJSxGKl

    P5LnyiDgDmNvueZY66bM2zqlZPgd5bkp3pDJv6rZAoGAaP5e5F1j6s82Pm7dCpH3

    mRZWnfZIKqoNQIq2BO8vA9/WrdFI2C27uNhxCp2ZDMulRdBZcoeHcwJjnyDzg4I4

    gBZ2nSKkVdlN1REoTjLBBdlHi8XkiXzxvpItc2wjNC2AKHaJqj/dnh3bbTAQD1iU

    AxPmmLJYYkhfZ2i1IrTVxZE =

    -----END RSA PRIVATE KEY-----

    KEYDATA;

    // sha256 + Pkcs1 <https://tools.ietf.org/html/rfc3447#page-43>

    $data = pack(‘H\*’, ‘3031300d060960864801650304020105000420’) . hash(‘sha256’, $data, true);

    $pk  = openssl\_get\_privatekey($key);

    openssl\_private\_encrypt($data, $res, $pk);

    return base64\_encode($res);

}

Data message in php:

$sign = $this->computeSignature(json\_encode($data, JSON\_UNESCAPED\_UNICODE));

$headers = [

    ‘Accept: application/json’,

    ‘Content-Type: application/json’,

    ‘X-Signature: ‘.$sign

];

$ch = new Curl();

$ch->init();

$ch->setOptions($x=[

    CURLOPT\_HEADER => false,

    CURLOPT\_SSL\_VERIFYPEER => false,

    CURLOPT\_SSLCERT => $this->client\_crt(),

    CURLOPT\_SSLKEY => $this->client\_key(),

    CURLOPT\_SSLCERTPASSWD => ‘1234’,

    CURLOPT\_HTTPHEADER => $headers,

]);

$result = $ch->post(

    ‘https://apip.orangedata.ru:2443/api/v2/documents/’,

    json\_encode($data, JSON\_UNESCAPED\_UNICODE));

Signature generation in C# using standard class library .net core:

private string ComputeSignature(string document)

{

    var data = Encoding.UTF8.GetBytes(document);

    using (var rsa = RSA.Create())

    {

        rsa.FromXmlString(privateKey);

        return Convert.ToBase64String(rsa.SignData(data, HashAlgorithmName.SHA256, RSASignaturePadding.Pkcs1));

    }

}

An example of signature generation in C # using the standard class library .net framework full:

private string ComputeSignature(string document)

{

    var data = Encoding.UTF8.GetBytes(document);

    using (var rsa = new RSACryptoServiceProvider())

    {

        rsa.FromXmlString(privateKey);

        return Convert.ToBase64String(rsa.SignData(data, “SHA256”));

    }

}

## 1.4. Testing API with Postman and Signature Generator application.

You can test OrangeData API with Postman application. Request body signing can be performed by Signature Generator application.  
<https://github.com/orangedata-official/API/tree/master/SignatureGenerator>

# 2. Requests

## 2.1 Receipt creation

Receipt generation is performed in 3 stages:

|  |  |
| --- | --- |
| **Stage 1:** | Client’s application (web site) sends an asynchronous request for receipt generation. Server stores the request data into processing queue. |
| **Stage 2:** | Server takes requests from the processing queue one-by-one and creates the corresponding fiscal document. Fiscal documents are created by fiscal registers, connected to the client’s account in the fiscal cloud. The resulting fiscal parameters are stored into database. |
| **Stage 3:** | Client’s application requests status of fiscal document and fiscal parameters assigned to the document. |

Each asynchronous request contain a “self-generated” document identifier (id field in the query). This field mast be unique within the organization. The id field is used for request status monitoring in stage 3 (see Section 2.2.).

A client can register any number of fiscal devices in [www.orangedata.ru](http://www.orangedata.ru) private zone. The number of devices determines the quantity of receipts that the client can generate each second. Maximum performance of a single caching device – is one receipt per second. If a company needs more receipts per second – more caching devices are required.

All caching devices of each company are combined into groups. Each device is categorized to one group in in [www.orangedata.ru](http://www.orangedata.ru) private zone. Groups allow clients to use specific device (or group of devices for receipt generation): if an initial request contain group name – the devices from specified group will be used for receipt generation. If group is not specified or NULL – the default group is used for receipt generation. The default group is “Main” for most clients or “vend” for vending clients or “transp” for transport companies (these three types of companies has special types of private zones in [www.orangedata.ru](http://www.orangedata.ru) and some specific in business-processes).

The system provides load-balancing of caching devices and selects “first free” matching device (with proper INN and proper group) for receipt generation.

Depending on the type of Personal Account, the default group name may differ:

|  |  |
| --- | --- |
| Personal account type | Default group name |
| Standard | Main |
| Vending | Vend |
| Transport | Transp or Courier \* |

\* The group will depend on what settings will be selected during the fiscalization of the cash register in the Orange Data Personal Account.

Automatic devices - Transp group

Manual Payment Terminals - Courier Group

Orange Data service stores requests for receipt generation for no more than 30 days. Long-term "storage of requests in the queue" may arise, for example, if the client has specified a group of cash registers in the request parameters (the group parameter), but no cash register is "tied" to this group in the personal account, or if the client has only one cash register and it " got up "due to blocking by the fiscal accumulator or for other reasons. If the request "has been in the queue" for more than 30 calendar days, it will not be executed.

After "punching the check", the result of the transaction is stored in the database for another 3 months. After 3 months, data from the database is moved to the archive.

Request: **POST** **/api/v2/documents/**

### 2.1.1 Request Body

|  |  |  |
| --- | --- | --- |
| Id | Document identifier | String from 1 to 64 symbols. Mandatory. |
| Inn | INN of the receipt receiving organization | String from 10 or 12 symbols. Mandatory. |
| Group | Group of devices that will be used to print receipt | String from 1 to 32 symbols or null |
| Content | Document contents | Structure Section 2.1.1.1 |
| Key | The name of the key used to verify the signature.  INN is used for clients,  code with prefix 301\*\*\*\* is used for partners  code with prefix 401\*\*\*\* is used for vending | String from 1 to 32 symbols. Mandatory. |
| callbackUrl | URL for sending the results of processing a receipt with a POST request | String from 1 to 1024 symbols or null |
| meta | Request metadata | String from 1 to 128 characters or null |
| ignoreItemCodeCheck | A flag indicating whether to ignore the CM check.  If the flag is not specified, then all CMs must successfully pass the check in order to generate a receipt: in tag 2106, bits number 0, 1, 2, 3 have the state "1"  If the flag is not specified and the CM has not passed the check, the check will not be generated and the status request will return the 422 Unprocessable Entity status | Boolean, optional.  The default is false. |

### 2.1.1.1 Document contents

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Description** | **Comment** | **FFD 1.05** | **FFD 1.2** |
| ffdVersion | FFD version number, 1209  2 - FFD 1.05, default value  4 - FFD 1.2 | Number, optional | Yes | Yes |
| Type | Settlement Type, tag 1054:   1. Income 2. Income return 3. Expense 4. Expense return | Number. Mandatory. | Yes | Yes |
| Positions | Settlement subject list, tag 1059 | Array of structures Section.2.1.1.2 | Yes | Yes |
| checkClose | Closing parameters of the receipt | Structure Section 2.1.1.3 | Yes | Yes |
| customerContact | Customer Phone Number or Email, tag 1008 | String from 1 to 64 symbols, format +{N} or {X}@{X}. Mandatory. | Yes | Yes |
| agentType | Agent Type, tag 1057. **Bytes** field, where bit number indicates that the user providing the service to the buyer (client) is:  0(1) – bank payment agent  1(2) – bank payment subagent  2(4) – payment agent  3(8) – payment subagent  4(16) – agent  5(32) – commission agent  6(64) – other agent  *The Receipt (accountable form) can contain the "agent marker" details (tag 1057) only if the registration report and / or the current re-registration report contains the "agent attribute" detail (tag 1057), which has a value identical to the value of the "agent attribute" detail (tag 1057) of the cash receipt.* | Number from 1 to 127, optional field | Yes | No |
| paymentTransferOperatorPhoneNumbers | Transfer Operator Phone Number, tag 1075 | Array of strings from 1 to 19 symbols, format +{N}, optional field | Yes | No |
| paymentAgentOperation | Payment Agent Operation, tag 1044 | String from 1 to 24, optional field | Yes | No |
| paymentAgentPhoneNumbers | Payment Agent Phone Number, tag 1073 | Array of strings from 1 to 19 symbols, format +{N}, optional field | Yes | No |
| paymentOperatorPhoneNumbers | Payment Operator Phone Number, tag 1074 | Array of strings from 1 to 19 symbols, format +{N}, optional field | Yes | No |
| paymentOperatorName | Payment Operator Name, tag 1026 | String from 1 to 64 symbols, optional field | Yes | No |
| paymentOperatorAddress | Payment Operator Address, tag 1005 | String from 1 to 243 symbols, optional field | Yes | No |
| paymentOperatorINN | Payment Operator INN, tag 1016 | String from 10 to 12 symbols, format NNNNNNNNNN, optional field | Yes | No |
| supplierPhoneNumbers | Supplier Phone Number, tag 1171 | Array of strings from 1 to 19 characters long, format +{N}, optional field | Yes | No |
| additionalUserAttribute | Additional User Attribute, tag 1084 | Structure Section 2.1.1.5, optional field | Yes | Yes |
| additionalAttribute | Additional Attribute (*accountable form*), tag 1192 | String from 1 to 16 symbols, optional field | Yes | Yes |
| automatNumber | Automat Number, tag 1036 | A string with a length of 1 to 20 characters, mandatory, if the group has a sign of transferring machine data, otherwise it should not be transmitted. **Only for vending and transport.** | Yes | Yes |
| settlementAddress | Settlement Address, tag 1009 | A string with a length of 1 to 243 characters, mandatory, if the group has a sign of machine data transmission, otherwise it should not be transmitted. **For vending, transport and couriers.** | Yes | Yes |
| settlementPlace | Settlement Place, tag 1187 | A string with a length of 1 to 243 characters, mandatory, if the group has a sign of machine data transmission, otherwise it should not be transmitted. **For vending, transport and couriers.** | Yes | Yes |
| Customer | Customer (client), tag 1227 | String from 1 to 243 characters, optional field | Yes | No |
| customerINN | Customer (client ) INN, tag 1228 | A string with a length of 10 to 12 characters, the format is NNNNNNNNN, optional field | Yes | No |
| Cashier | Cashier, tag 1021 | String from 1 to 64 characters, optional field **Mandatory for couriers** | Yes | Yes |
| cashierINN | Cashier INN, tag 1203 | 12 character string format NNNNNNNNNNNN, optional field | Yes | Yes |
| senderEmail | Sender Email, tag 1117 | String from 1 to 64 characters, optional field | Yes | Yes |
| totalSum | Total Sum (*accountable form*), tag 1020.  Tax Authority allows rounding in this field. The Tax Authority requirements are: «*Tag 1020 is calculated as sum of all sums for each positions in a receipt (tags 1043). If tag 1020 is calculated by external calculation device, the value of tag 1020 without kopeks mast be the equal to sum of all positions in the receipt without kopeks»*.  OrangeData company recommends clients to use exact value in this field and do not apply rounding to avoid confusions. | Decimal number up to 2 characters after the dot, optional field | Yes | Yes |
| vat1Sum | The amount of the VAT receipt at a rate of 20%, tag 1102 | Decimal number up to 2 characters after the dot, optional field | Yes | Yes |
| vat2Sum | The amount of the VAT receipt at a rate of 10%, tag 1103 | Decimal number up to 2 characters after the dot, optional field | Yes | Yes |
| vat3Sum | Settlement amount with VAT at a rate of 0%, tag 1104 | Decimal number up to 2 characters after the dot, optional field | Yes | Yes |
| vat4Sum | Settlement amount per receipt without VAT, tag 1105 | Decimal number up to 2 characters after the dot, optional field | Yes | Yes |
| vat5Sum | VAT amount of the receipt at the settlement rate of 20/120, tag 1106 | Decimal number up to 2 characters after the dot, optional field | Yes | Yes |
| vat6Sum | VAT amount of the receipt at the settlement rate of 10/110, tag 1107 | Decimal number up to 2 characters after the dot, optional field | Yes | Yes |
| customerInfo | Customer details, 1256 | Structure p.2.1.1.8, optional field | No | Yes |
| operationalAttribute | Operational requisite of the check, 1270 | Structure 2.1.1.9, optional field | No | Yes |
| industryAttribute | Industry check requisite, 1261 | Structure 2.1.1.10, optional field | No | Yes |

Note: Due to historical reasons, the numeration of fields **vat1Sum, vat2Sum, vat3Sum, vat4Sum, vat5Sum, vat6Sum** in OrangeData formats is different from tax rates numeration in Fiscal Formats tag 1199. Use this guide strictly when you develop an integration solution.  
  
For transport / vending, one of the required parameters are tags (**1036, 1009, 1187**)

For couriers, one of the required parameters are tags (**1021, 1009, 1187**)

### 2.1.1.2 Subject of Settlement

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Description** | **Comment** | **FFD 1.05** | **FFD 1.2** |
| Quantity | Quantity of the subject of settlement, tag 1023 | Decimal number up to 6 characters after the dot\*. Mandatory. | Yes | Yes |
| Price | Price per unit of the settlement subject, including discounts and extra charges, tag 1079 | Decimal number up to 2 characters after the dot \*. Mandatory. | Yes | Yes |
| Tax | VAT rate, 1199:  1 – VAT rate 20%  2 – VAT rate 10%  3 – VAT rate calculation 20/120  4 – VAT rate calculation 10/110  5 – VAT rate 0%  6 – VAT is not charged | Number from 1 to 6. Mandatory. | Yes | Yes |
| taxSum | The amount of VAT per item, tag 1200  *This parameter is relevant for VAT rates 1-4. For bids 5 and 6, this tag in the subject of calculation is not transmitted, the transmitted value is ignored.*  *If you pass the value 0, then the tag will not be written.* | Decimal number up to 2 characters after the dot, optional field | Yes | Yes |
| Text | Name of the settlement item, tag 1030 | String up to 128 symbols. Mandatory. | Yes | Yes |
| paymentMethodType | Payment Method Type, tag 1214:  1 – Prepayment 100%  2 – Partial prepayment  3 – Prepaid expense  4 – Full settlement  5 – Partial settlement and credit  6 – Transfer on credit  7 – loan payment | Number from 1 to 7 or null. If null is passed, then the value of 4 will be transferred to the subject of settlement. Full settlement. | Yes | Yes |
| paymentSubjectType | Payment Subject Type, tag 1212:  1 – Product  2 – Excisable goods  3 – Work  4 – Service  5 – Gamble bet  6 – Gambling win  7 – Lottery ticket  8 – Lottery win  9 – Providing Intellectual Property  10 – Payment  11 – Agent Fee  12 – Payoff  13 – Other subject of settlement  14 – Property right  15 – Non-operating income \*  16 – Other payments and contributions \*  17 - Sales tax  18 - Resort fee  19 – Pledge  20 – Consumption  21 – Contributions to compulsory pension insurance for individual entrepreneurs  22 – Contributions to compulsory pension insurance  23 – Contributions to compulsory health insurance for individual entrepreneurs  24 – Contributions for compulsory health insurance  25 – Contributions to compulsory social security  26 – Casino payment | A number from 1 to 26 or null. If null is passed, then the value of 1 will be transferred to the subject of settlement. Product.  For values 15 and 16, see the note. | Yes | Yes |
| nomenclatureCode | Nomenclature Code, tag 1162 | String containing **base64** encoded array from 8 to 32 bytes or null | Yes | No |
| supplierInfo | Supplier Info, tag 1224 | Structure Section 2.1.1.6, or null | Yes | Yes |
| supplierINN | Supplier INN, tag 1226 | String 10 to 12 characters long, optional field | Yes | Yes |
| agentType | Agent Type, 1222. **Bytes** field, where bit number indicates that the user providing the service to the buyer (client) is:  0(1) – bank payment agent  1(2) – bank payment subagent  2(4) – payment agent  3(8) – payment subagent  4(16) – agent  5(32) – commission agent  6(64) – other agent | Number from 1 to 127, optional field | Yes | Yes |
| agentInfo | Agent Info, tag 1223 | Structure Section 2.1.1.7, or null. 243 bytes max. | Yes | Yes |
| unitOfMeasurement | Unit Of Measurement, tag 1197 | String from 1 to 16 symbols, optional field | Yes | No |
| additionalAttribute | Additional Attribute of Subject Settlement, tag 1191 | String from 1 to 64 symbols, optional field | Yes | Yes |
| manufacturerCountryCode | Manufacturer Country Code, tag 1230 | A string with a length of 1 to 3 characters, format NNN, an optional field.  The service will automatically pad the string with up to 3 characters with spaces. | Yes | Yes |
| customsDeclarationNumber | Customs Declaration Number, tag 1231 | String from 1 to 32 symbols, optional field | Yes | Yes |
| Excise | Excise, tag 1229 | Decimal up to 2 symbols after dot, optional field | Yes | Yes |
| unitTaxSum | The amount of VAT per unit of the subject of calculation, 1198 | Decimal number with an accuracy of 2 characters after the dot, optional field | Yes | Yes |
| quantityMeasurementUnit | Measure of the amount of the subject of calculation, 2108 | Number from 0 to 255, if not specified | No | Yes |
| ItemCode | Product code, 1163 | Structure 2.1.1.11, optional field | No | Yes |
| fractionalQuantity | Fractional quantity of labeled goods, 1291 | Structure 2.1.1.12, optional field | No | Yes |
| industryAttribute | Sectoral requisite of the subject of calculation, 1260 | Structure 2.1.1.10, optional field | No | Yes |

Note:

1. In case the requisite “settlement item” (tag 1059) of the receipt contains the requisite “settlement item type” (tag 1212), having the value “15”, then the receipt must be transferred to the OFD in electronic form and the requisite “settlement item name "(tag 1030) must contain one of the values from" 1 " to" 25 "specified in Table 2.1.1.2.1.

2. If the item “settlement item” (tag 1059) contains the attribute “settlement item type” (tag 1212) with the value “16” as part of the receipt, then the receipt must be transferred to the OFD in electronic form and the attribute “settlement item name "(tag 1030) must contain one of the values from " 26 "to" 31 "specified in table 2.1.1.2.1.

2.1.1.2.1 Settlement Item Name

|  |  |  |
| --- | --- | --- |
| The value of the requisite | The "name of the subject of calculation" (tag 1030) contains information | Format |
| "1" | income from equity participation in other organizations, except for income used to pay for additional shares of shares placed among shareholders organizations | income from equity participation in other organizations |
| "2" | income in the form of positive (negative) exchange rate difference generated due to deviation of the rate of sale (purchase) of foreign currency from the official rate set by the Central Bank of the Russian Federation on the date of transfer of ownership of foreign currency (the specifics of defining the banks ' profits from these transactions are established by article 290 of the Tax code of the Russian Federation) | income in the form of exchange differences resulting from the deviation of the foreign currency sale (purchase) rate from the official exchange rate |
| "3" | income in the form of fines, penalties and (or) other sanctions recognized by the debtor or payable by the debtor on the basis of a court decision that has entered into force, as well as amounts of compensation for losses or damage | income in the form of fines, penalties and (or) other penalties payable by the debtor for breach of contractual obligations |
| "4" | income from the lease of property (including land plots), if such income is not determined by the taxpayer in accordance with the procedure established by article 249 of the Tax code of the Russian Federation of Contractual obligations | income from renting out property (including land plots) (sublease) |
| "5" | income from the use of rights to the results of intellectual activity and rights to equated means of individualization (in particular, from the use of rights arising from patents for inventions, utility models, industrial designs), if such income is not determined by the taxpayer in the manner prescribed by article 249 of the Tax code of the Russian Federation | income from the granting of rights to the results of intellectual activity |
| "6" | income in the form of interest received under loan agreements, loans, Bank accounts, Bank deposits, as well as securities and other debt obligations (the specifics of determining banks ' income in the form of interest are established by article 290 of the Tax code of the Russian Federation) | income from interest received under loan agreements and other debt obligations |
| "7" | income in the form of amounts of restored reserves, the expenses for the formation of which were accepted as part of expenses in the manner and under the conditions established by the articles 266, 267, 267.2, 267.4, 292, 294, 294.1, 297.3, 300, 324 and 324.1 of the Tax code of the Russian Federation | income from amounts of restored reserves |
| "8" | income in the form of gratuitously received property (works, services) or property rights, except for the cases specified in article 251 of the Tax code of the Russian Federation | income from gratuitously received property (works, services) or property rights |
| "9" | income in the form of income distributed in favor of the taxpayer with his participation in a simple partnership, accounted for in accordance with article 278 of the Tax code of the Russian Federation | income from distributed in favor of the taxpayer with his participation in a simple partnership |
| "10" | income in the form of income from previous years identified in the reporting (tax) period | income from income from previous years identified in the reporting (tax) period |
| "11" | income in the form of a positive exchange rate difference, except for a positive exchange rate difference arising from the revaluation of advances issued (received | income from of a positive exchange rate difference |
| "12" | income in the form of fixed assets and intangible assets received free of charge in accordance with international agreements of the Russian Federation or with the legislation of the Russian Federation by nuclear power plants to improve their safety, used for non-production purposes | income from of fixed assets and intangible assets received by nuclear power plants free of charge |
| "13" | income in the form of the cost of received materials or other property during dismantling or disassembly during the liquidation of decommissioned fixed assets (except for cases provided for in subparagraph 18 of paragraph 1 of article 251 of the Tax code of the Russian Federation) | income in the form of the cost of materials received during the liquidation of decommissioned fixed assets |
| "14" | income in the form of non-intended use of property (including funds), works, services received as part of charitable activities (including charitable assistance, donations), targeted revenues, and targeted funding, with the exception of budget funds. The budget legislation of the Russian Federation applies to budget funds that are not used for their intended purpose | income in the form of property, works, and services that were not used for their intended purpose |
| "15" | income in the form of funds that are not used for their intended purpose by enterprises and organizations that include particularly radiation-hazardous and nuclear-hazardous productions and facilities, intended for forming reserves to ensure the safety of these productions and facilities at all stages of their life cycle and development in accordance with the legislation of the Russian Federation on the use of nuclear energy | income in the form of funds that were not used for the intended purpose, intended for the formation of reserves to ensure the safety of production |
| "16" | The income amounts to which the reporting (tax) period there was a reduction in the authorized (share) capital (Fund) of the organization if such a reduction effected with a simultaneous refusal to refund the relevant part of the contributions (deposits) shareholders (participants) of the organization (with the exception of cases provided by the subparagraph 17 of item 1 of article 251 of the Tax code of the Russian Federation) | income in the form of amounts by which the authorized (share) capital (Fund) of the organization is reduced |
| "17" | income in the form of refund amounts from a non-profit organization of previously paid contributions (contributions) if such contributions (contributions) were previously included in expenses when forming the tax base | income in the form of refund amounts from a non-profit organization of previously paid contributions (deposits) |
| "18" | income in the form of amounts of accounts payable (obligations to creditors) written off in connection with the expiration of the Statute of limitations or on other grounds, except for the cases provided for in subparagraph 21 of paragraph 1 of article 251 of the Tax code of the Russian Federation. The provisions of this paragraph shall not apply to the cancellation of a mortgage agent of payables in the form of liabilities to holders of mortgage-backed bonds, as well as on cancellation of a specialized society payable in the form of obligations to holders of issued bonds | income in the form of amounts of accounts payable written off due to the expiration of the Statute of limitations or for other reasons |
| "19" | income in the form of income received from operations with derivative financial instruments, subject to the provisions of articles 301-305 of the Tax code of the Russian Federation | income in the form of income received from operations with derivative financial instruments |
| "20" | income in the form of the value of surplus inventory and other assets that are identified as a result of inventory | income in the form of the value of surplus inventory and other assets that are identified as a result of inventory |
| "21" | income in the form of the cost of mass media products and book products that are subject to replacement when returning or writing off such products on the grounds provided for in subparagraphs 43 and 44 of paragraph 1 of article 264 of the Tax code of the Russian Federation. The cost of the products specified in this paragraph is estimated in accordance with the procedure for evaluating the balance of finished products established by article 319 of the Tax code of the Russian Federation | income in the form of the cost of media products and book products to be replaced when returning or debiting |
| "22" | income in the form of amounts of adjustment of the taxpayer's profit due to the application of methods for determining for tax purposes the compliance of prices used in transactions with market prices (profitability) provided for in articles 105.12 and 105.13 of the Tax code of the Russian Federation | income in the form of tax payer's profit adjustment amounts |
| "23" | income is returned to the donor or his assigns monetary equivalent of immovable property and (or) securities transferred for the replenishment of the target capital of the noncommercial organization in the order established by the Federal law of December 30, 2006 of N 275-FZ "On procedure of formation and utilization of target capital of the noncommercial organisations" (meeting of the legislation of the Russian Federation, 2006, N 1 (h 1), article 38; 2013, N 30 (h. 1), article 4084) | income in the form of the returned monetary equivalent of real estate and (or) securities transferred to replenish the endowment capital of a non-profit organization |
| "24" | income in the form of the difference between the amount of tax deductions from excise amounts accrued when performing operations specified in subparagraphs 21, 23-33 of paragraph 1 of article 182 of the Tax code of the Russian Federation and the specified excise amounts | income in the form of the difference between the amount of tax deductions from excise amounts and the specified excise amounts |
| "25" | income in the form of profit of a controlled foreign company determined in accordance with the Tax code of the Russian Federation - for organizations recognized in accordance with the Tax code of the Russian Federation as controlling persons of this foreign company | income in the form of profit of a controlled foreign company |
| "26" | insurance premiums for mandatory pension insurance | Insurance premiums for mandatory pension insurance |
| "27" | insurance premiums for compulsory social insurance in the event of temporary disability and maternity | insurance premiums for compulsory social insurance in the event of temporary disability and maternity |
| "28" | insurance premiums for compulsory medical insurance | Insurance premiums for mandatory medical insurance |
| "29" | insurance premiums for compulsory social insurance against industrial accidents and occupational diseases | insurance premiums for compulsory social insurance against industrial accidents and occupational diseases |
| "30" | the payment in accordance with the legislation of the Russian Federation of temporary disability allowance (except for accidents at work and occupational illnesses) for days of temporary incapacity of the employee that are paid for by the employer and whose number is established by the Federal law from December, 29th, 2006 N 255-FZ "On compulsory social insurance against temporary disability and in connection with motherhood" (meeting of the legislation of the Russian Federation, 2007, N 1 (part 1), art 18; 2018, N 11, art. 1591), in the part not covered by insurance payments made to employees by insurance organizations that have licenses issued in accordance with the legislation of the Russian Federation for the implementation of the relevant type of activity, under contracts with employers in favor of employees in the event of their temporary disability (except for accidents at work and occupational diseases) for the days of temporary disability, which are paid at the expense of the employer and the number of which is established by the Federal law of December 29, 2006 N 255-FZ "on compulsory social insurance in the event of temporary disability and in connection with maternity" | the temporary disability benefits |
| "31" | payments (contributions) under voluntary personal insurance contracts concluded with insurance organizations that have licenses issued in accordance with the legislation of the Russian Federation for the implementation of the relevant type of activity, in favor of employees in the event of their temporary disability (except for accidents at work and occupational diseases) for the days of temporary disability, which are paid at the expense of the employer and the number of which is established by the Federal law of December 29, 2006 N 255-FZ "on compulsory social insurance in the event of temporary disability and in connection with maternity". | payments for voluntary personal insurance |

### 2.1.1.3 Receipt Close Parameters

|  |  |  |
| --- | --- | --- |
| payments | payments | Array of structures |
| taxationSystem | Taxation system, 1055:  0 – General Taxation System  1 – Simplified revenue  2 – Simplified revenue minus expense  3 – Unified tax on imputed income  4 – Uniform agricultural tax  5 – Patent system of taxation | Number from 0 to 5 |

Note:

Attention! From 01.01.2021 UTII is canceled. To select the appropriate tax regime, you can use a special [service](https://www.nalog.gov.ru/rn77/service/mp/).

### 2.1.1.4 Payment

|  |  |  |
| --- | --- | --- |
| Type | Payment type:  1 – amount of the receipt in cash, 1031  2 – the amount of the receipt is non-cash, 1081  14 the amount of the receipt prepaid (offset of the advance and (or) previous payments), 1215  15 – the amount of the check postpaid (on credit),1216  16 – amount on the receipt (accountable form) by counter-provision 1217 | A number from 1 to 16. Mandatory. |
| amount | Payment amount | Decimal number with an accuracy of 2 characters after the dot\*. Mandatory. |

\* The maximum amount of the receipt is 99,999,999. 99 rubles. The maximum amount of the quantity \* price position after rounding is also 99,999,999. 99 rubles. The maximum price per unit of the settlement item is 99,999,999.99 rubles, the maximum number of the settlement item is 281,474,976,710,655.

### 2.1.1.5 Additional user details

|  |  |  |
| --- | --- | --- |
| name | Name of additional user details, 1085 | String from 1 to 64 characters.  The length of the name + value attributes must not exceed 235 characters in total. |
| value | Value of additional user details, 1086 | String from 1 to 234 characters.  The length of the name + value attributes must not exceed 235 characters in total. |

### 2.1.1.6 Supplier Data

|  |  |  |
| --- | --- | --- |
| phoneNumbers | Supplier's phone number, 1171 | Array of strings from 1 to 19 characters in length, format +{N}, optional field |
| name | Name of the supplier, 1225 | String of up to 239 characters.  Note: *the 243-character data includes the supplier's phone numbers + 4 characters for each phone.*  For example, if you send 2 vendor phones with 12 and 14 characters in length, the maximum length of the vendor name will be 239 – (12 + 4) – (14 + 4) = 205 characters |

### 2.1.1.7 Agent Data

|  |  |  |
| --- | --- | --- |
| paymentTransferOperatorPhoneNumbers | Phone number of the transfer operator, 1075 | Array of strings from 1 to 19 characters in length, format +{N}, optional field |
| paymentAgentOperation | The operation of the paying agent, 1044 | String length from 1 to 24 characters, optional field |
| paymentAgentPhoneNumbers | Phone number of the payment agent,1073 | Array of strings from 1 to 19 characters in length, format +{N}, optional field |
| paymentOperatorPhoneNumbers | Payment Operator Phone Numbers, 1074 | Array of strings from 1 to 19 characters in length, format +{N}, optional field |
| paymentOperatorName | Payment Operator Name, 1026 | String length from 1 to 64 characters, optional field |
| paymentOperatorAddress | Payment Operator Address, 1005 | String length from 1 to 243 characters, optional field |
| paymentOperatorINN | Transfer operator's INN, 1016 | A string of 10 to 12 characters in length, in the format of NNNNNNNNNN, optional field |

### 2.1.1.8 Customer (customer) information, 1256

|  |  |  |
| --- | --- | --- |
| name | Buyer (client), 1227 | String from 1 to 239 characters, optional |
| inn | TIN of the buyer (client), 1228 | String from 10 to 12 characters long, format ЦЦЦЦЦЦЦЦЦЦЦЦ ЦЦЦЦЦЦЦЦЦЦЦЦ, an optional field |
| birthDate | Date of birth of the buyer (client), 1243 | String 10 characters, format DD.MM.YYYY, optional field |
| citizenship | Citizenship, 1244 | A string of 1 to 3 characters in length, CTSC format, optional field. |
| identityDocumentCode | Identity document type code, 1245 | The service will automatically complete the string with spaces up to 3 characters. |
| address | Buyer (client) address, 1254 | String 2 characters long, format ЦЦ, optional field |

### 2.1.1.9 Operational details of the check, 1270

|  |  |  |
| --- | --- | --- |
| date | Date, time of operation, 1273 | Time as string in ISO8601 format, optional field |
| id | Operation ID, 1271  *Accepts the values "0" until the value of the FTS of Russia requisite is determined.* | Number from 0 to 255, optional field, optional field |
| value | Operation data, 1272 | String from 1 to 64 characters, optional field |

### 2.1.1.10 Industry check requisite, 1261

|  |  |  |
| --- | --- | --- |
| foivId | FOIV ID, 1262 | String from 1 to 3 characters, optional |
| causeDocumentDate | Date of foundation document, 1263 | String 10 characters, format DD.MM.YYYY, optional field |
| causeDocumentNumber | Foundation document number, 1264 | String from 1 to 32 characters, optional |
| value | Industry props value, 1265 | String from 1 to 239 characters, optional |

### 2.1.1.11 Product code, 1163

|  |  |  |
| --- | --- | --- |
| urecognized | unrecognized product code, 1300  *The value of the variable consists of a sequence of characters read from a bar code of an unidentified type and not identified as one of the attributes "КТ EAN-8" (tag 1301), "КТ EAN-13" (tag 1302), "КТ ITF-14" (tag 1303), "KT GS1.0" (tag 1304), "KT GS1.M" (tag 1305), "KT KMK" (tag 1306), "KT MI" (tag 1307), "KT EGAIS-2.0 "(Tag 1308)," CT EGAIS-3.0 "(tag 1309)," CT F.1 "(tag 1320)," CT F.2 "(tag 1321)," CT F.3 "(tag 1322), "CT F.4" (tag 1323), "CT F.5" (tag 1324), "CT F.6" (tag 1325).*  *If the length of the sequence exceeds 32 characters, then the sequence is truncated to 32 characters from the left* | String from 1 to 32 characters, optional |
| ean8 | CT EAN-8, 1301 | 8 character string, optional |
| ean13 | CT EAN-13, 1302 | String 13 characters, optional |
| itf14 | CT ITF-14, 1303 | String 14 characters, optional |
| gs1 | CT GS1.0, 1304 | String from 1 to 38 characters, optional |
| gs1m | CT GS1.M, 1305 | String from 1 to 38 characters, optional |
| kmk | KT KMK, 1306 | String from 1 to 38 characters, optional |
| mi | CT MI, 1307 | String 20 characters, optional |
| egais20 | CT EGAIS-2.0, 1308 | String 33 characters, optional |
| egais30 | CT EGAIS-3.0, 1309 | String 14 characters, optional |
| f1 | CT F.1, 1320 | String from 1 to 32 characters, optional |
| f2 | CT F.2, 1321 | String from 1 to 32 characters, optional |
| f3 | CT F.3, 1322 | String from 1 to 32 characters, optional |
| f4 | CT F.4, 1323 | String from 1 to 32 characters, optional |
| f5 | CT F.5, 1324 | String from 1 to 32 characters, optional |
| f6 | CT F.6, 1325 | String from 1 to 32 characters, optional |

### 2.1.1.12 Fractional quantity of labeled goods, 1291

|  |  |  |
| --- | --- | --- |
| Numerator | Numerator, 1293 | Unsigned integer 8 bytes |
| Denominator | Denominator, 1294 | Unsigned integer 8 bytes |

Response: API can return the following HTTP status codes

* 201 Created – the check was created and added to the processing queue, and the response body is empty
* 401 Unauthorized client certificate failed verification
* 409 Conflict– receipt with this ID has already been created in the system, and the response body is empty
* 400 Bad Request – the transmitted data contains validation errors, or the signature failed validation, response body section 2.1.2
* 503, Service Unavailable – the document queue is full, the response returns the Retry-After header with a timeout in seconds, after which it is necessary to repeat the request, the body of the response item 2. 1. 2.

Request Example:

{

  "id": "12345678990",

  "inn": "123456789012",

  "group": "Main",

  "content": {

    "type": 1,

    "positions": [

      {

        "quantity": 1.000,

        "price": 123.45,

        "tax": 6,

        "text": "Булка",

        "paymentMethodType": 4,

        "paymentSubjectType": 1

      },

      {

        "quantity": 2.000,

        "price": 4.45,

        "tax": 4,

        "text": "Спички",

"paymentMethodType": 3,

        "paymentSubjectType": 13

      }

    ],

    "checkClose": {

      "payments": [

        {

          "type": 1,

          "amount": 123.45

        },

        {

          "type": 2,

          "amount": 8.90000

        }

      ],

      "taxationSystem": 1

    },

    "customerContact": "foo@example.com"

  }

}

Example of a request with agent data, additional user details, supplier data, machine number, settlement address, and settlement place:

{

  "id": "12345678990",

  "inn": "123456789012",

  "group": "Main",

  "key": "1234567",

  "content": {

    "type": 1,

    "positions": [

      {

        "quantity": 1.000,

        "price": 123.45,

        "tax": 6,

        "text": "Булка",

        "paymentMethodType": 4,

        "paymentSubjectType": 1,

        "nomenclatureCode": "igQVAAADMTIzNDU2Nzg5MDEyMwAAAAAAAQ==",

        "agentType": 127,

        "agentInfo": {

          "paymentTransferOperatorPhoneNumbers": [ "+79200000001", "+74997870001" ],

          "paymentAgentOperation": "Какая-то операция 1",

          "paymentAgentPhoneNumbers": [ "+79200000003" ],

          "paymentOperatorPhoneNumbers": [ "+79200000002", "+74997870002" ],

          "paymentOperatorName": "ООО \"Атлант\"",

          "paymentOperatorAddress": "Воронеж, ул. Недогонная, д. 84",

          "paymentOperatorINN": "7727257386"

        },

        "unitOfMeasurement": "Кг",

        "additionalAttribute": "Доп. атрибут и все тут",

        "manufacturerCountryCode": "643",

        "customsDeclarationNumber": "АД 11/77 от 01.08.2018",

        "excise": 23.45

      },

      {

        "quantity": 2.000,

        "price": 4.45,

        "tax": 4,

        "text": "Спички",

        "paymentMethodType": 3,

        "paymentSubjectType": 13,

        "supplierINN": "9715225506",

        "supplierInfo": {

          "phoneNumbers": [ "+79266660011", "+79266660022" ],

          "name": "ПАО \"Адамас\""

        }

      }

    ],

    "checkClose": {

      "payments": [

        {

          "type": 1,

          "amount": 123.45

        },

        {

          "type": 2,

          "amount": 8.90000

        }

      ],

      "taxationSystem": 1

    },

    "customerContact": "foo@example.com",

    "agentType": 127,

    "paymentTransferOperatorPhoneNumbers": [ "+79260000001", "+74957870001" ],

    "paymentAgentOperation": "Какая-то операция",

    "paymentAgentPhoneNumbers": [ "+79260000003" ],

    "paymentOperatorPhoneNumbers": [ "+79260000002", "+74957870002" ],

    "paymentOperatorName": "ООО \"Росинка\"",

    "paymentOperatorAddress": "Москва, Мастеркова 4",

    "paymentOperatorINN": "9715225506",

    "supplierPhoneNumbers": [ "+74957870004" ],

    "additionalUserAttribute": {

      "name": "Любимая цитата",

      "value": "В здоровом теле здоровый дух, этот лозунг еще не потух!"

    },

    "automatNumber": "123456789",

    "settlementAddress": "г.Москва, Красная площадь, д.1",

    "settlementPlace": "Палата №6",

    "additionalAttribute": "Доп атрибут чека",

    "customer": "Кузнецов Иван Петрович",

    "customerINN": "789456123488"

  }

}

An example of a request generated in accordance with FFD 1.2:

{

  "id": "12345678990",

  "inn": "7725327863",

  "group": "Main",

  "key": "1234567",

  "content": {

    "type": 1,

    "positions": [

      {

        "quantity": 1.000,

        "price": 123.45,

        "tax": 6,

        "text": "Булка",

        "paymentMethodType": 4,

        "paymentSubjectType": 1,

        "agentType": 127,

        "agentInfo": {

          "paymentTransferOperatorPhoneNumbers": [ "+79200000001", "+74997870001" ],

          "paymentAgentOperation": "Какая-то операция 1",

          "paymentAgentPhoneNumbers": [ "+79200000003" ],

          "paymentOperatorPhoneNumbers": [ "+79200000002", "+74997870002" ],

          "paymentOperatorName": "ООО \"Атлант\"",

          "paymentOperatorAddress": "Воронеж, ул. Недогонная, д. 84",

          "paymentOperatorINN": "7727257386"

        },

        "additionalAttribute": "Доп. атрибут и все тут",

        "manufacturerCountryCode": "643",

        "customsDeclarationNumber": "АД 11/77 от 01.08.2018",

        "excise": 23.45,

        "unitTaxSum": 0.23,

        "itemCode": "010460406000600021N4N57RSCBUZTQ\u001d2403004002910161218\u001d1724010191ffd0\u001d92tIAF/YVoU4roQS3M/m4z78yFq0fc/WsSmLeX5QkF/YVWwy8IMYAeiQ91Xa2z/fFSJcOkb2N+uUUmfr4n0mOX0Q==",

        "plannedStatus": 2,

        "fractionalQuantity": {

          "numerator": 1,

          "denominator": 2

        },

        "industryAttribute": {

          "foivId": "012",

          "causeDocumentDate": "12.08.2021",

          "causeDocumentNumber": "666",

          "value": "position industry"

        },

        "barcodes": {

          "ean8": "46198532",

          "ean13": "4006670128002",

          "itf14": "14601234567890",

          "gs1": "010460043993125621JgXJ5.T",

          "mi": "RU-401301-AAA0277031",

          "egais20": "NU5DBKYDOT17ID980726019",

          "egais30": "13622200005881",

          "f1": null,

          "f2": null,

          "f3": null,

          "f4": null,

          "f5": null,

          "f6": null

        }

      },

      {

        "quantity": 2.000,

        "price": 4.45,

        "tax": 4,

        "text": "Спички",

        "paymentMethodType": 3,

        "paymentSubjectType": 13,

        "supplierINN": "9715225506",

        "supplierInfo": {

          "phoneNumbers": [ "+79266660011", "+79266660022" ],

          "name": "ПАО \"Адамас\""

        },

        "quantityMeasurementUnit": 10

      }

    ],

    "checkClose": {

      "payments": [

        {

          "type": 1,

          "amount": 123.45

        },

        {

          "type": 2,

          "amount": 8.90000

        }

      ],

      "taxationSystem": 1

    },

    "customerContact": "foo@example.com",

    "additionalUserAttribute": {

      "name": "Любимая цитата",

      "value": "В здоровом теле здоровый дух, этот лозунг еще не потух!"

    },

    "automatNumber": "123456789",

    "settlementAddress": "г.Москва, Красная площадь, д.1",

    "settlementPlace": "Палата №6",

    "additionalAttribute": "Доп атрибут чека",

    "cashier": "Кассир",

    "senderEmail": "ru@example.mail",

    "customerInfo": {

      "name": "Кузнецов Иван Петрович",

      "inn": "7725327863",

      "birthDate": "15.09.1988",

      "citizenship": "643",

      "identityDocumentCode": "01",

      "identityDocumentData": "multipassport",

      "address": "Басеенная 36"

    },

    "operationalAttribute": {

      "date": "2021-08-12T18:36:16",

      "id": 0,

      "value": "operational"

    },

    "industryAttribute": {

      "foivId": "010",

      "causeDocumentDate": "11.08.2021",

      "causeDocumentNumber": "999",

      "value": "industry"

    },

    "ffdVersion": 4

  },

  "meta": "some meta",

  "callbackUrl": "http://call.back/?doc=2",

  "ignoreItemCodeCheck": false

}

### 2.1.2 Response body

|  |  |  |
| --- | --- | --- |
| errors | Array of request processing errors | Array of strings |

Example of an error response:

{

  "errors": [

    "Не указан идентификатор документа 'Id'",

    "Не указан ИНН организации 'INN'",

    "Отсутствует содержимое документа 'Content'"

  ]

}

## 2.2 Receipt status

Request: **GET** **/api/v2/documents/****{inn}/status****/{document\_id}**

**{inn}** – INN of the organization for which the receipt is being punched

**{document\_id}** – ID of the document that was specified when it was created

The SHA256-RSA signature is not used in this request.

Response codes:

* 202 Accepted – the receipt was created and added to the queue for processing, but has not yet been processed, the response body is empty
* 400 Bad Request – the organization was not found, and the receipt with the specified ID was not found
* 401 Unauthorized the client certificate failed validation
* 200 OK – the receipt is processed, the response body 2.2.1
* 524, Document Expired Before Processing – the server was unable to process the document within the allotted time, empty response body. Send the receipt again with a new identifier for re-processing.
* 422, Unprocessable Entity - the document cannot be processed due to CM validation errors, the response body is clause 2.2.2

### 2.2.1 Response body

|  |  |  |
| --- | --- | --- |
| id | ID (this field contain the same data that were in the field id in the initial request – see Section 2.1) | String from 1 to 64 characters |
| sequentialId | Sequential unique id of document within all comnanie’s fiscal documents. This is mandatory field if option “CountDocuments” is true. Default setting  “CountDocuments”=false and most clients do not use this field. | Number |
| deviceSN | Factory number of the device that printed the receipt | String up to 20 characters |
| deviceRN | Registration number of the device that printed the receipt | String up to 20 characters |
| fsNumber | Fiscal storage serial number | String of 16 characters |
| ofdName | Name of OFD | String up to 256 characters |
| ofdWebsite | Website of OFD | String up to 58? symbols' |
| ofdINN | IIN of OFD | String of 12 characters |
| fnsWebsite | Website of federal taxation service | String up to 256 characters |
| companyINN | INN company (this field contain the same data that were in the field inn in the initial request – see Section 2.1) | String of 12 characters |
| companyName | Company Name | String up to 256 characters |
| documentNumber | Document Number | Number |
| shiftNumber | Shift Number | Number |
| documentIndex | The check number for a shift | Number |
| processedAt | Time of registration of the fiscal document in the Federal tax service | Time as a string in ISO8601 format |
| content | Document content | Structure of section 2. 1. 1. 1 |
| change | change | Decimal number up to 2 characters after the dot |
| fp | Fiscal sign | String of 10 characters |
| callbackUrl | URL for sending the results of processing a receipt with a POST request | String from 1 to 1024 characters or null |
| meta | Query metadata | String 1 to 128 characters or null |

Response example:

{

  "id": "12345678990",

  "deviceSN": "0000000000001358",

  "deviceRN": "0000000400054952",

  "fsNumber": "9999078900001341",

  "ofdName": "ООО \"Ярус\"(\"ОФД-Я\")",

  "ofdWebsite": "www.ofd-ya.ru",

  "ofdinn": "7728699517",

  "fnsWebsite": "www.nalog.ru",

  "companyINN": "123456789012",

  "companyName": "ЗАО ТОРГОВЫЙ ОБЪЕКТ №1",

  "documentNumber": 117,

  "shiftNumber": 20,

  "documentIndex": 5,

  "processedAt": "2017-02-14T14:16:00",

  "content": {

    "type": 1,

    "positions": [

      {

        "quantity": 1.000,

        "price": 123.45,

        "tax": 6,

        "text": "Булка",

        "paymentMethodType": 4,

        "paymentSubjectType": 1

      },

      {

        "quantity": 2.000,

        "price": 4.45,

        "tax": 4,

        "text": "Спички",

        "paymentMethodType": 3,

        "paymentSubjectType": 13

      }

    ],

    "checkClose": {

      "payments": [

        {

          "type": 1,

          "amount": 123.45

        },

        {

          "type": 2,

          "amount": 8.90000

        }

      ],

      "taxationSystem": 1

    },

    "customerContact": "+79123456789"

  },

  "change": 974.01,

  "fp": "2364009522"

}

### **2.2.2 Response body with CM processing errors**

|  |  |  |
| --- | --- | --- |
| errors | Array of request processing errors | Array of structures in clause 2.2.2.1 |

### 2.2.2.1 Error processing the subject of calculation containing CM

|  |  |  |
| --- | --- | --- |
| position | Position number starting from 0 | Number 2 bytes |
| itemCode | Marking code, 2000 | String from 1 to 243 characters |
| checkResult | Result of checking | Structure of clause 2.2.2.2 |

## 2.2.2.2 Result of CM check

|  |  |  |
| --- | --- | --- |
| fsCheckStatus | KM check result, 2004 | A number from 0 to 255 |
| fsCheckStatusCause | The reason why the local check was not carried out:  0 - KM is checked in FN  1 - CM of this type is not subject to verification in the FN  2 - FN does not contain a key for checking the verification code of this CM  3 - Verification not possible because GS1 Application Identifiers 91 and / or 92 are missing or not formatted correctly.  4 - Checking CM in FN is impossible for another reason | A number from 0 to 255 |
| fsItemCodeType | Marking code type, 2100 | A number from 0 to 255 |
| checkResult | Product details check result, 2106 | A number from 0 to 255 |
| oismCheckResultCode | FN response code to the online check command  If 0x20, then the reason is returned in the next byte:  1 - Wrong fiscal response attribute;  2 - Invalid format of the response details;  3 - Invalid request number in the response;  4 - Invalid FN number;  5 - Invalid data block CRC;  7 - Invalid response length.  The value is 0xFF if the server did not respond within the timeout. | A number from 0 to 255 |
| oismResponse | OISM response to the request.  The field is absent if oismCheckResultCode has a value other than 0 or the checkout is offline. | Structure clause 2.2.2.3, optional field |

### 2.2.2.3 Server response to CM check request

|  |  |  |
| --- | --- | --- |
| dateTime | Date and time of request, 2114 | Time as a string in ISO8601 format |
| resultCode | Request handling codes, 2105 | A number from 0 to 255 |
| itemCodeType | Marking code type, 2100 | Number from 0 to 255, optional |
| oismStatus | OISM response on the status of goods, 2109 | Number from 0 to 255, optional |
| processingResults | Query processing results, 2005 | Number from 0 to 255, optional |
| itemId | Item ID, 2101 | String from 1 to 38 characters, optional |

Sample response:

[

  {

    "position": 0,

    "itemCode": "123",

    "checkResult": {

      "fsCheckStatus": 2,

      "fsCheckStatusCause": 3,

      "fsItemCodeType": 4,

      "checkResult": 1,

      "oismCheckResultCode": 5,

      "oismResponse": {

        "resultCode": 1,

        "itemCodeType": 2

      }

    }

  }

]

## 2.3 Creating a correction receipt

Creating a correction receipt – the asynchronous request, after its execution, a receipt is queued for handling. In addition to the receipt data, the client sends a unique document ID using which, in the future, it will be able to request the status of the correction receipt (p. 2. 4). In addition, this ID provides idempotency of requests.

The ID must be unique within the organization.

The INN is used to determine the device that can be used to make a receipt.

Group can be null or omitted in the request, in which case the receipt will be sent to the default group.

The default group is the group named Main.

Request: **POST** **/api/v2/****corrections/**

### 2.3.1 Request body

|  |  |  |
| --- | --- | --- |
| id | id | String from 1 to 64 characters. This parameter is required. |
| Inn | INN of the organization for which the receipt is being printing | String of 10 or 12 characters. This parameter is required. |
| group | Group of devices that will be used to do the receipt | String from 1 to 32 characters or null |
| content | Content | Structure of clause 2. 3. 1. 1 |
| key | The name of the key used to verify the signature.  INN is used for clients,  code with prefix 301\*\*\*\* is used for partners  code with prefix 401\*\*\*\* is used for vending | String of 1 to 32 characters long, or null. This parameter is required. |
| callbackUrl | URL for sending the results of processing a receipt with a POST request | String from 1 to 1024 characters or null |
| meta | Query metadata | String 1 to 128 characters or null |

### 2.3.1.1 Document content

|  |  |  |
| --- | --- | --- |
| correctionType | Correction type 1173:  0. Independently  1. According to the prescription | Number. This parameter is required. |
| type | The basis of calculation, 1054:  1. Coming  3. Expenditure | Number. This parameter is required. |
| causeDocumentDate | Date of the basis document for correction 1178.  In this detail, the time is always specified as 00:00:00 | Time as a string in ISO 8601 format. This parameter is required. |
| causeDocumentNumber | The number of Tax Authority prescription on the basis of which the current correction operation is made, 1179  See the field correctionType (tag 1173) | String from 1 to 32 characters. This parameter is required. |
| totalSum | he amount of the calculation specified in the receipt, 1020 | Decimal number with an accuracy of 2 characters after the dot. This parameter is required. |
| cashSum | The amount of the check (for cash), 1031 | Decimal number up to 2 characters after the dot |
| eCashSum | The amount of the check (non-cash), 1081 | Decimal number up to 2 characters after the dot |
| prepaymentSum | he amount of the check by prepayment (offset of the advance and (or) previous payments), 1215 | Decimal number up to 2 characters after the dot |
| postpaymentSum | The amount of the check by prepayment (offset of the advance and (or) previous payments), 1215 | Decimal number up to 2 characters after the dot |
| otherPaymentTypeSum | The amount of the check counter-provision, 1217 | Decimal number up to 2 characters after the dot |
| tax1Sum | VAT check amount at the rate of 20%, 1102 | Decimal number up to 2 characters after the dot |
| tax2Sum | VAT check amount at the rate of 10%, 1103 | Decimal number up to 2 characters after the dot |
| tax3Sum | The amount of calculation on the receipt with VAT at the rate of 0%, 1104 | Decimal number up to 2 characters after the dot |
| tax4Sum | Payment amount for the receipt without VAT, 1105 | Decimal number up to 2 characters after the dot |
| tax5Sum | The VAT amount of the receipt is calculated. bid 20/120, 1106 | Decimal number up to 2 characters after the dot |
| tax6Sum | The VAT amount of the receipt is calculated. the rate of 10/110, 1107 | Decimal number up to 2 characters after the dot |
| taxationSystem | Taxation system, 1055:  0 – General Taxation System  1 – Simplified revenue  2 – Simplified revenue minus expense  3 – Unified tax on imputed income  4 – Uniform agricultural tax  5 – Patent system of taxation | Number |
| automatNumber | Automat Number, 1036 | A string from 1 to 20 characters long, a mandatory field if the group has the automatic data transfer feature, otherwise it should not be passed. **Only for vending and transport.** |
| settlementAddress | Settlement Address, 1009 | A string between 1 and 243 characters long, a mandatory field if the group has the automatic data transfer feature, otherwise it should not be passed. **Only for vending and transport.** |
| settlementPlace | Settlement Place, 1187 | A string between 1 and 243 characters long, a mandatory field if the group has the automatic data transfer feature, otherwise it should not be passed. **Only for vending and transport.** |

Response: API can return the following HTTP status codes

* 201 Created – the receipt was created and added to the processing queue, and the response body is empty
* 401 Unauthorized client certificate failed verification
* 409 Conflict– receipt with this ID has already been created in the system, and the response body is empty
* 400 Bad Request – the transmitted data contains validation errors, or the signature failed validation, response body section 2.1.2
* 503, Service Unavailable – the document queue is full, the response returns the Retry-After header with a timeout in seconds, after which it is necessary to repeat the request, the body of the response item 2. 1. 2.

### 2.3.2 Response body with request processing errors

|  |  |  |
| --- | --- | --- |
| errors | Array of request processing errors | Array of strings |

Request Example:

{

  "id": "12345678990",

  "inn": "123456789012",

  "group": "Main",

  "content": {

    "correctionType": 1,

    "type": 1,

    "causeDocumentDate": "2017-08-10T00:00:00",

    "causeDocumentNumber": "ФЗ-54",

    "totalSum": 17.25,

    "cashSum": 1.23,

    "eCashSum": 2.34,

    "prepaymentSum": 5.67,

    "postpaymentSum": 4.56,

    "otherPaymentTypeSum": 3.45,

    "tax1Sum": 1.34,

    "tax2Sum": 2.34,

    "tax3Sum": 3.34,

    "tax4Sum": 4.34,

    "tax5Sum": 5.34,

    "tax6Sum": 6.34,

    "taxationSystem": 1,

    "automatNumber": "123456789",

    "settlementAddress": "г.Москва, Красная площадь, д.1",

    "settlementPlace": "Палата №6",

  }

}

Error example:

{

  "errors": [

    "Не указан идентификатор документа 'Id'",

    "Не указан ИНН организации 'INN'",

    "Отсутствует содержимое документа 'Content'"

  ]

}

## 2.4 Correction Receipt Status

Request: **GET** **/api/v2/corrections/{inn}/status/{document\_id}**

**{inn}** – Organization INN

**{document\_id}** – ID of the document that was specified when it was created

The SHA256-RSA signature is not used in this request.

Response: API can return the following status codes

* 202 Accepted – receipt created and added to the processing queue, but not yet processed, empty response body
* 400 Bad Request – organization not found, the receipt with the specified ID was not found
* 401 Unauthorized – client certificate failed verification
* 200 OK – the receipt has been processed

### 2.4.1 Response Body

|  |  |  |
| --- | --- | --- |
| id | Document identifier | String from 1 to 64 symbols |
| deviceSN | The serial number of the device that printed the receipt | String up to 20 symbols |
| deviceRN | Registration number of the device that printed the receipt | String up to 20 symbols |
| fsNumber | Fiscal Storage Number | String of 16 symbols |
| ofdName | OFD Name | String up to 256 symbols |
| ofdWebsite | OFD Website | String up to 58? symbols |
| ofdINN | OFD INN | String of 12 symbols |
| fnsWebsite | FNS Website | String up to 256 symbols |
| companyINN | Company INN | String of 12 symbols |
| companyName | Company Name | String up to 256 symbols |
| documentNumber | Document Number | Number |
| shiftNumber | Shift Number | Number |
| documentIndex | Shift Receipt Number | Number |
| processedAt | Fiscal document registration time | Time in the string format ISO8601 |
| content | Content | Structure Section 2.3.1.1 |
| fp | Fiscal indication | String of 10 symbols |
| callbackUrl | URL for sending the results of processing a POST receipt request | String from 1 to 1024 symbols or null |
| meta | Query metadata | String 1 to 128 characters or null |

Response Example:

{

  "id": "12345678990",

  "deviceSN": "0000000000001358",

  "deviceRN": "0000000400054952",

  "fsNumber": "9999078900001341",

  "ofdName": "ООО \"Ярус\"(\"ОФД-Я\")",

  "ofdWebsite": "www.ofd-ya.ru",

  "ofdinn": "7728699517",

  "fnsWebsite": "www.nalog.ru",

  "companyINN": "123456789012",

  "companyName": "ЗАО ТОРГОВЫЙ ОБЪЕКТ №1",

  "documentNumber": 117,

  "shiftNumber": 20,

  "documentIndex": 5,

  "processedAt": "2017-02-14T14:16:00",

  "content": {

    "correctionType": 1,

    "type": 1,

    "description": "НЕ ХОЧЕТСЯ НО НАДО",

    "causeDocumentDate": "2017-08-10T00:00:00",

    "causeDocumentNumber": "ФЗ-54",

    "totalSum": 17.25,

    "cashSum": 1.23,

    "eCashSum": 2.34,

    "prepaymentSum": 5.67,

    "postpaymentSum": 4.56,

    "otherPaymentTypeSum": 3.45,

    "tax1Sum": 1.34,

    "tax2Sum": 2.34,

    "tax3Sum": 3.34,

    "tax4Sum": 4.34,

    "tax5Sum": 5.34,

    "tax6Sum": 6.34,

    "taxationSystem": 1

  },

  "change": 974.01,

  "fp": "2364009522"

}

## 2.5 Cashing machine status in the group

Request: **GET** **/api/v2/****devices/status/{inn}/{group\_name}**

**{inn}** – Organization INN

**{ group\_name }** – device group name

The SHA256-RSA signature is not used in this request.

Response: api can return the following status codes

* 400 Bad Request – organization or group not found, response body Section 2.5.2
* 401 Unauthorized – client certificate failed verification
* 200 OK – no errors, response body Section.2.5.1

### 2.5.1 Response Body

|  |  |  |
| --- | --- | --- |
| devices | Devices | Array of Structures Section.2.5.1.1 |

### 2.5.1.1 The structure of the object information about the device

|  |  |  |
| --- | --- | --- |
| deviceSN | Device Serial Number | String up to 20 symbols |
| fsNumber | Fiscal Storage Number | String 16 symols, optional field |
| ofdName | OFD Name | String up to 256 symbols, optional field |
| ofdAddress | Address/IP and port for sending checks to the OFD | String up to 64 symbols, optional field |
| unsentDocumentsCount | Unsent Documents Count | Number |
| firstUnsentDocumentDate | First Unsent Document Date | Time format ISO8601, optional field |
| fsDocumentsCount | Total number of documents in FS | Number |
| fsExpirationDate | Expiration Date of FS | Time format ISO8601, optional field |

### 2.5.2 Request processing errors response body

|  |  |  |
| --- | --- | --- |
| errors | Array of request processing errors | Array of strings |

Response:

{

  "devices": [

    {

      "deviceSN": "0000000000001358",

      "fsNumber": "9999078900001341",

      "ofdName": "ООО \"Ярус\"(\"ОФД-Я\")",

      "ofdAddress": "192.168.1.31:1215",

      "unsentDocumentsCount": 42,

      "firstUnsentDocumentDate": "2019-12-09T14:16:00",

      "fsDocumentsCount": 666,

      "fsExpirationDate": "2020-12-09T14:16:00"

    },

    {

      "deviceSN": "0000000000001359",

      "fsNumber": "9999078900001342",

      "ofdName": null,

      "ofdAddress": null,

      "unsentDocumentsCount": 0,

      "firstUnsentDocumentDate": null,

      "fsDocumentsCount": 0,

      "fsExpirationDate": null

    }

  ]

}

### 2.6 Checking access parameters

The request is used to check the access parameters: certificate and signature.

Request: POST / api / v2 / check /

### 2.6.1 Request body

|  |  |  |
| --- | --- | --- |
| inn | INN of the organization for which the check is being issued | A string of 10 or 12 characters. This parameter is required. |
| group | A group of devices with which the check will be punched | String from 1 to 32 characters or null |
| key | The name of the key to be used to verify the signature. For customers, their INN is used, for partners and paying agents, a code with a mask is 301 \*\*\*\*, for vending 401 \*\*\*\*. | String from 1 to 32 characters. This parameter is required. |

In response: the api can return the following http status codes

• 200 OK - request completed successfully

• 401 Unauthorized - the client certificate did not pass verification

• 400 Bad Request - the data has an incorrect format or the signature did not pass verification, the response body is clause 2.4.2

### 2.6.2 Response body with request processing errors

|  |  |  |
| --- | --- | --- |
| errors | Array of request processing errors | Array of strings |

Request example:

{

  "inn": "123456789012",

  "group": "Main",

  "key": "666"

}

An example of an error response:

{

  "errors": [

    "Подпись не прошла проверку"

  ]

}

### 2.7 Creating a check for FFD correction 1.2

Request: POST / api / v2 / correction12 /

### 2.7.1 Request body

|  |  |  |
| --- | --- | --- |
| id | Document ID | String from 1 to 64 characters. This parameter is required. |
| inn | INN of the organization for which the check is being issued | A string of 10 or 12 characters. This parameter is required. |
| group | A group of devices with the help of which the check will be punched | String from 1 to 32 characters or null |
| content | Document content | Structure of clause 2.7.1.1 |
| key | The name of the key to be used to verify the signature. For customers, their INN is used, for partners and paying agents a code with a mask is 301 \*\*\*\*, for vending 401 \*\*\*\*. | String from 1 to 32 characters. This parameter is required. |
| callbackUrl | URL for sending the results of processing a check with a POST request | String from 1 to 1024 characters or null |
| meta | Request metadata | String from 1 to 128 characters or null |
| ignoreItemCodeCheck | A flag indicating whether to ignore the CM check.  If the flag is not specified, then all CMs must successfully pass the check in order to generate a receipt: in tag 2106, bits number 0, 1, 2, 3 have the state "1"  If the flag is not specified and the CM has not passed the check, the check will not be generated and the status request will return the 422 Unprocessable Entity status | Boolean, optional. The default is false. |

### 2.7.1.1 Document content

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Description** | **Comment** |
| ffdVersion | FFD version number, 1209  2 - FFD 1.05, default value  4 - FFD 1.2 | Number, optional |
| type | Calculation attribute, 1054:  1. Parish  2. Return of the receipt  3. Consumption  4. Return of expense | Number. This parameter is required. |
| positions | List of items of calculation, 1059 | Array of structures in clause 2.1.1.2 |
| checkClose | Check closing options | Structure of clause 2.1.1.3 |
| customerContact | Buyer's phone or email address, 1008 | String from 1 to 64 characters, format + {D} or {C} @ {C}. This parameter is required. |
| correctionType | Correction type 1173:  0. Independently  1. By prescription | Number. This parameter is required. |
| causeDocumentDate | The document date of the basis for the correction is 1178.  In this requisite, always indicate the time as 00:00:00 | Time as a string in ISO8601 format. This parameter is required. |
| causeDocumentNumber | The number of the prescription of the tax authority, on the basis of which this correction is carried out, 1179 | String from 1 to 32 characters. This parameter is required. |
| additionalUserAttribute | Additional user requisite, 1084 | Structure p.2.1.1.5, optional field |
| additionalAttribute | Additional check details (BSO), 1192 | String from 1 to 16 characters, optional field |
| automatNumber | Vending machine number, 1036 | A string from 1 to 20 characters long, a required field if the group has the sign of the machine's data transfer, otherwise it should not be transferred. **For vending and transport only**. |
| settlementAddress | Settlement address, 1009 | A string with a length of 1 to 243 characters, a required field if the group has the sign of the machine's data transfer, otherwise it should not be transferred. **For vending and transport only.** |
| settlementPlace | Place of settlements, 1187 | A string with a length of 1 to 243 characters, a required field if the group has the sign of the machine's data transfer, otherwise it should not be transferred. **For vending and transport only.** |
| cashier | Cashier, 1021 | String from 1 to 64 characters, optional field |
| cashierINN | INN cashier, 1203 | A string of 12 characters, format ЦЦЦЦЦЦЦЦЦЦЦЦ, optional |
| senderEmail | Check sender email address, 1117 | String from 1 to 64 characters, optional |
| totalSum | The amount of the calculation indicated in the check (SRF) is 1020.  *Fiscal Document formats allow rounding down in this field. Literally, the FFD requirement reads as follows: "The value in rubles, excluding kopecks, should be equal to the value of the sum of all the details" the cost of the subject of calculation, taking into account discounts and margins "(tag 1043) in rubles, excluding kopecks."*  To avoid inconsistencies, we recommend using an exact value without rounding. | Decimal number up to 2 characters after the period, optional |
| vat1Sum | The amount of VAT check at the rate of 20%, 1102 | Decimal number up to 2 characters after the dot, optional |
| vat2Sum | The amount of VAT check at the rate of 10%, 1103 | Decimal number up to 2 characters after the dot, optional |
| vat3Sum | Calculation amount for a check with VAT at a rate of 0%, 1104 | Decimal number up to 2 characters after the period, optional |
| vat4Sum | Check settlement amount without VAT, 1105 | Decimal number up to 2 characters after the period, optional |
| vat5Sum | The amount of VAT on the check rate 20/120, 1106 | Decimal number up to 2 characters after the period, optional |
| vat6Sum | The amount of VAT on the check rate 10/110, 1107 | Decimal number up to 2 characters after the period, optional |
| customerInfo | Customer (customer) details, 1256 | Structure p.2.1.1.8, optional field |
| industryAttribute | Industry check requisite, 1261 | Structure p.2.1.1.10, optional field |

**Note:** for historical reasons, the numbering of tax rates vat1Sum, vat2Sum, vat3Sum, vat4Sum, vat5Sum, vat6Sum in OrangeData formats is slightly different from the numbering of tax rates in FFD for tag 1199. When developing integration, you should strictly adhere to this guide.

In response: the api can return the following http status codes

• 201 Created - a receipt has been created and added to the queue for processing, an empty response body  
 • 401 Unauthorized - the client certificate did not pass verification  
 • 409 Conflict - a receipt with this identifier has already been created in the system, an empty response body  
 • 400 Bad Request - the transmitted data contains validation errors, or the signature did not pass verification, the response body is clause 2.1.2  
 • 503, Service Unavailable - the document queue is full, the response returns a Retry-After header with a timeout in seconds, after which it is worth repeating the request, response body p. 2.7.2.

Request example:

{

  "id": "12345678990",

  "inn": "7725327863",

  "group": "Main",

  "key": "1234567",

  "content": {

    "type": 1,

    "positions": [

      {

        "quantity": 1.000,

        "price": 123.45,

        "tax": 6,

        "text": "Булка",

        "paymentMethodType": 4,

        "paymentSubjectType": 1,

        "agentType": 127,

        "agentInfo": {

          "paymentTransferOperatorPhoneNumbers": [ "+79200000001", "+74997870001" ],

          "paymentAgentOperation": "Какая-то операция 1",

          "paymentAgentPhoneNumbers": [ "+79200000003" ],

          "paymentOperatorPhoneNumbers": [ "+79200000002", "+74997870002" ],

          "paymentOperatorName": "ООО \"Атлант\"",

          "paymentOperatorAddress": "Воронеж, ул. Недогонная, д. 84",

          "paymentOperatorINN": "7727257386"

        },

        "additionalAttribute": "Доп. атрибут и все тут",

        "manufacturerCountryCode": "643",

        "customsDeclarationNumber": "АД 11/77 от 01.08.2018",

        "excise": 23.45,

        "unitTaxSum": 0.23,

        "itemCode": "010460406000600021N4N57RSCBUZTQ\u001d2403004002910161218\u001d1724010191ffd0\u001d92tIAF/YVoU4roQS3M/m4z78yFq0fc/WsSmLeX5QkF/YVWwy8IMYAeiQ91Xa2z/fFSJcOkb2N+uUUmfr4n0mOX0Q==",

        "plannedStatus": 2,

        "fractionalQuantity": {

          "numerator": 1,

          "denominator": 2

        },

        "industryAttribute": {

          "foivId": "012",

          "causeDocumentDate": "12.08.2021",

          "causeDocumentNumber": "666",

          "value": "position industry"

        },

        "barcodes": {

          "ean8": "46198532",

          "ean13": "4006670128002",

          "itf14": "14601234567890",

          "gs1": "010460043993125621JgXJ5.T",

          "mi": "RU-401301-AAA0277031",

          "egais20": "NU5DBKYDOT17ID980726019",

          "egais30": "13622200005881"

        }

      },

      {

        "quantity": 2.000,

        "price": 4.45,

        "tax": 4,

        "text": "Спички",

        "paymentMethodType": 3,

        "paymentSubjectType": 13,

        "supplierINN": "9715225506",

        "supplierInfo": {

          "phoneNumbers": [ "+79266660011", "+79266660022" ],

          "name": "ПАО \"Адамас\""

        },

        "quantityMeasurementUnit": 10

      }

    ],

    "checkClose": {

      "payments": [

        {

          "type": 1,

          "amount": 123.45

        },

        {

          "type": 2,

          "amount": 8.90000

        }

      ],

      "taxationSystem": 1

    },

    "customerContact": "foo@example.com",

    "correctionType": 1,

    "causeDocumentDate": "2021-08-13T00:00:00",

    "causeDocumentNumber": "42/666",

    "additionalUserAttribute": {

      "name": "Любимая цитата",

      "value": "В здоровом теле здоровый дух, этот лозунг еще не потух!"

    },

    "automatNumber": "123456789",

    "settlementAddress": "г.Москва, Красная площадь, д.1",

    "settlementPlace": "Палата №6",

    "additionalAttribute": "Доп атрибут чека",

    "cashier": "Кассир",

    "senderEmail": "ru@example.mail",

    "customerInfo": {

      "name": "Кузнецов Иван Петрович",

      "inn": "7725327863",

      "birthDate": "15.09.1988",

      "citizenship": "643",

      "identityDocumentCode": "01",

      "identityDocumentData": "multipassport",

      "address": "Басеенная 36"

    },

    "operationalAttribute": {

      "date": "2021-08-12T18:36:16",

      "id": 0,

      "value": "operational"

    },

    "industryAttribute": {

      "foivId": "010",

      "causeDocumentDate": "11.08.2021",

      "causeDocumentNumber": "999",

      "value": "industry"

    },

    "ffdVersion": 4

  },

  "meta": "some meta",

  "callbackUrl": "http://call.back/?doc=2",

  "ignoreItemCodeCheck": false

}

### 2.7.2 Response body with request processing errors

|  |  |  |
| --- | --- | --- |
| errors | Array of request processing errors | Array of strings |

An example of an error response:

{

  "errors": [

    "Не указан идентификатор документа 'Id'",

    "Не указан ИНН организации 'INN'",

    "Отсутствует содержимое документа 'Content'"

  ]

}

### 2.8 Status of FFD correction receipt 1.2

Request: **GET / api / v2 / correction12 / {inn} / status / {document\_id}**

**{inn}** - INN of the organization for which the check is being issued

**{document\_id}** - document identifier that was specified when it was created

This request does not use the SHA256-RSA signature.

Answer: api can return the following status codes

• 202 Accepted - the receipt was created and added to the queue for processing, but not yet processed, empty response body

• 400 Bad Request - the organization was not found, the receipt with the specified identifier was not found

• 401 Unauthorized - the client certificate did not pass verification

• 200 OK - check processed, response body p.2.8.1

• 422, Unprocessable Entity - the document cannot be processed due to CM validation errors, the response body is clause 2.2.2

### 2.8.1 Response body

|  |  |  |
| --- | --- | --- |
| id | Document identifier | String from 1 to 64 symbols |
| deviceSN | The serial number of the device that printed the receipt | String up to 20 symbols |
| deviceRN | Registration number of the device that printed the receipt | String up to 20 symbols |
| fsNumber | Fiscal Storage Number | String of 16 symbols |
| ofdName | OFD Name | String up to 256 symbols |
| ofdWebsite | OFD Website | String up to 58? symbols |
| ofdINN | OFD INN | String of 12 symbols |
| fnsWebsite | FNS Website | String up to 256 symbols |
| companyINN | Company INN | String of 12 symbols |
| companyName | Company Name | String up to 256 symbols |
| documentNumber | Document Number | Number |
| shiftNumber | Shift Number | Number |
| documentIndex | Shift Receipt Number | Number |
| processedAt | Fiscal document registration time | Time in the string format ISO8601 |
| content | Content | Structure Section 2.1.1.1 |
| fp | Fiscal indication | String of 10 symbols |
| callbackUrl | URL for sending the results of processing a POST receipt request | String from 1 to 1024 symbols or null |
| meta | Query metadata | String 1 to 128 characters or null |

Request example:

{

  "id": "12345678990",

  "deviceSN": "0000000000001358",

  "deviceRN": "0000000400054952",

  "fsNumber": "9999078900001341",

  "ofdName": "ООО \"Ярус\"(\"ОФД-Я\")",

  "ofdWebsite": "www.ofd-ya.ru",

  "ofdinn": "7728699517",

  "fnsWebsite": "www.nalog.ru",

  "companyINN": "123456789012",

  "companyName": "ЗАО ТОРГОВЫЙ ОБЪЕКТ №1",

  "documentNumber": 117,

  "shiftNumber": 20,

  "documentIndex": 5,

  "processedAt": "2017-02-14T14:16:00",

  "content": {

    "type": 1,

    "positions": [

      {

        "quantity": 1.000,

        "price": 123.45,

        "tax": 6,

        "text": "Булка",

        "paymentMethodType": 4,

        "paymentSubjectType": 1

      },

      {

        "quantity": 2.000,

        "price": 4.45,

        "tax": 4,

        "text": "Спички",

        "paymentMethodType": 3,

        "paymentSubjectType": 13

      }

    ],

    "checkClose": {

      "payments": [

        {

          "type": 1,

          "amount": 123.45

        },

        {

          "type": 2,

          "amount": 8.90000

        }

      ],

      "taxationSystem": 1

    },

    "customerContact": "+79123456789"

  },

  "change": 974.01,

  "fp": "2364009522"

}

### 

### 2.9 Checking the marking code

Request: POST / api / v2 / itemcode /

### 2.9.1 Request body

|  |  |  |
| --- | --- | --- |
| id | Document ID | String from 1 to 64 characters. This parameter is required. |
| inn | INN of the organization for which the request is being sent | A string of 10 or 12 characters. This parameter is required. |
| group | The group of devices with which the request will be processed | String from 1 to 32 characters or null |
| content | Document content | Structure of clause 2.9.1.1 |
| key | The name of the key to be used to verify the signature. For customers, their TIN is used, for partners and paying agents, a code with a mask is 301 \*\*\*\*, for vending 401 \*\*\*\*. | String from 1 to 32 characters. This parameter is required. |
| callbackUrl | URL to send the results of processing POST check request | String from 1 to 1024 characters or null |
| meta | Request metadata | String from 1 to 128 characters or null |

### 2.9.1.1 Document content

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Description** | **Comment** |
| plannedStatus | Planned status, 2003 | A number from 0 to 256 |
| ItemCode | Marking code, 2000  KM itself, as it was read by the scanner | String from 1 to 243 characters |
| quantityMeasurementUnit | Measure of the amount of the subject of calculation, 2108 | Number from 0 to 255, optional |
| quantity | Number of the subject of calculation, 1023 | Decimal number up to 6 characters after the period, optional. |
| fractionalQuantity | Fractional quantity of labeled goods, 1291 | Structure p.2.1.1.12, optional field |

In response: the api can return the following http status codes

• 201 Created - a receipt has been created and added to the queue for processing, an empty response body  
 • 401 Unauthorized - the client certificate did not pass verification  
 • 409 Conflict - a receipt with this identifier has already been created in the system, an empty response body  
 • 400 Bad Request - the transmitted data contains validation errors, or the signature did not pass verification, the response body is clause 2.9.2  
 • 503, Service Unavailable - the document queue is full, the response returns a Retry-After header with a timeout in seconds, after which it is worth repeating the request, response body p.2.9.2.

Request example:

{

  "plannedStatus": 1,

  "itemCode": "00000046210654eK7fYtcAAModGVz"

}

{

  "plannedStatus": 2,

  "itemCode": "00000046210654eK7fYtcAAModGVz",

  "quantityMeasurementUnit": 21,

  "quantity": 1.234

}

{

  "plannedStatus": 4,

  "itemCode": "00000046210654eK7fYtcAAModGVz",

  "quantityMeasurementUnit": 0,

  "quantity": 1,

  "fractionalQuantity": {

    "numerator": 1,

    "denominator": 2

  }

}

### 2.9.2 Response body with request processing errors

|  |  |  |
| --- | --- | --- |
| errors | Array of request processing errors | Array of strings |

An example of an error response:

{

  "errors": [

    "Не указан идентификатор документа 'Id'",

    "Не указан ИНН организации 'INN'",

    "Отсутствует содержимое документа 'Content'"

  ]

}

### 2.10 Marking code check status

Request: **GET / api / v2 / itemcode / {inn} / status / {document\_id}**

**{inn}** - TIN of the organization for which the check is being issued

**{document\_id}** - document identifier that was specified when it was created

This request does not use the SHA256-RSA signature.

Answer: api can return the following status codes

• 202 Accepted - the request has been created and added to the queue for processing, but has not been processed yet, the response body is empty

• 400 Bad Request - the organization was not found, the request with the specified identifier was not found

• 401 Unauthorized - the client certificate did not pass verification

• 200 OK - request processed, response body p.2.10.1

### 2.10.1 Response body

|  |  |  |
| --- | --- | --- |
| id | Document ID | String from 1 to 64 characters |
| deviceSN | Serial number of the device that processed the request | String up to 20 characters |
| deviceRN | Registration number of the device that processed the request | String up to 20 characters |
| fsNumber | Fiscal accumulator number | String 16 characters |
| processedAt | Server request processing time | Time as a string in ISO8601 format |
| content | Document content | Structure of clause 2.9.1.1 |
| checkResult | Result of checking | Structure p.2.2.2 |
| callbackUrl | URL for sending the results of processing a check with a POST request | String from 1 to 1024 characters or null |
| meta | Request metadata | String from 1 to 128 characters or null |

Request example:

{

  "id": "006f14f6d6d048c0859c0f0bafa1389c",

  "deviceSN": "0000000000001358",

  "deviceRN": "0000000400054952",

  "fsNumber": "9999078900001341",

  "processedAt": "2021-08-06T15:17:26",

  "content": {

    "plannedStatus": 2,

    "itemCode": "00000046210654eK7fYtcAAModGVz",

    "quantityMeasurementUnit": 0,

    "quantity": 2.0

  },

  "checkResult": {

    "fsCheckStatus": 2,

    "fsCheckStatusCause": 3,

    "fsItemCodeType": 4,

    "checkResult": 1,

    "oismCheckResultCode": 5,

    "oismResponse": {

      "resultCode": 1,

      "itemCodeType": 2

    }

  }

}

### 2.11 Status of FFD correction receipt 1.2

Request: GET / api / v2 / correction12 / {inn} / status / {document\_id}

{inn} - TIN of the organization for which the check is being issued

{document\_id} - document identifier that was specified when it was created

This request does not use the SHA256-RSA signature.

Answer: api can return the following status codes

• 202 Accepted - the receipt was created and added to the queue for processing, but not yet processed, empty response body

• 400 Bad Request - the organization was not found, the receipt with the specified identifier was not found

• 401 Unauthorized - the client certificate did not pass verification

• 200 OK - check processed, response body p.2.11.1

• 524, Document Expired Before Processing - the server failed to process the document within the allotted time, send a receipt with a new identifier for reprocessing, empty response body

• 422, Unprocessable Entity - the document cannot be processed due to CM validation errors, the response body is clause 2.2.2

### 2.11.1 Response body

|  |  |  |
| --- | --- | --- |
| id | Document ID | String from 1 to 64 characters |
| deviceSN | Serial number of the device that punched the check | String up to 20 characters |
| deviceRN | Registration number of the device that punched the check | String up to 20 characters |
| fsNumber | Fiscal accumulator number | String 16 characters |
| ofdName | OFD name | String up to 256 characters |
| odfWebsite | CRF website | Line up to 58? characters |
| odfINN | OFD INN | String 12 characters |
| fnsWebsite | FTS website | String up to 256 characters |
| companyINN | User's TIN | String 12 characters |
| companyName | User name | String up to 256 characters |
| documentNumber | FD number | Number |
| shiftNumber | Shift number | Number |
| documentIndex | Check number per shift | Number |
| processedAt | Time of registration of a fiscal document in the FN | Time as string in ISO8601 format |
| content | Document content | Structure of clause 2.7.1.1 |
| change | Change | Decimal number up to 2 characters after the dot |
| fp | Fiscal sign | String 10 characters |
| callbackUrl | URL for sending the results of processing a check with a POST request | String from 1 to 1024 characters or null |
| meta | Request metadata | String from 1 to 128 characters or null |

Sample answer:

{

  "id": "12345678990",

  "deviceSN": "0000000000001358",

  "deviceRN": "0000000400054952",

  "fsNumber": "9999078900001341",

  "ofdName": "ООО \"Ярус\"(\"ОФД-Я\")",

  "ofdWebsite": "www.ofd-ya.ru",

  "ofdinn": "7728699517",

  "fnsWebsite": "www.nalog.ru",

  "companyINN": "123456789012",

  "companyName": "ЗАО ТОРГОВЫЙ ОБЪЕКТ №1",

  "documentNumber": 117,

  "shiftNumber": 20,

  "documentIndex": 5,

  "processedAt": "2017-02-14T14:16:00",

  "content": {

    "type": 1,

    "correctionType": 1,

    "causeDocumentDate": "2017-08-10T00:00:00",

    "causeDocumentNumber": "ФЗ-54",

    "positions": [

      {

        "quantity": 1.000,

        "price": 123.45,

        "tax": 6,

        "text": "Булка",

        "paymentMethodType": 4,

        "paymentSubjectType": 1

      },

      {

        "quantity": 2.000,

        "price": 4.45,

        "tax": 4,

        "text": "Спички",

        "paymentMethodType": 3,

        "paymentSubjectType": 13

      }

    ],

    "checkClose": {

      "payments": [

        {

          "type": 1,

          "amount": 123.45

        },

        {

          "type": 2,

          "amount": 8.90000

        }

      ],

      "taxationSystem": 1

    },

    "customerContact": "+79123456789"

  },

  "change": 974.01,

  "fp": "2364009522"

}

# 3. Swagger

The api description is available as an Open API Specification (PAS).

Swagger-ui is available by **url / swagger**

Specification is available by url **/swagger/v2/swagger.json**

# 4. Links to the receipt.

This functionality is enabled in your personal account on the "Settings" page, under "Allow customers access to receipts".

As soon as the request for fiscal processing is processed by the cashier, this link will display the fiscal document in electronic form (receipt). Before that, there will be an inscription about processing the receipt.

The mask for generating: https://**{url}/{inn}**/**{document\_id}**

**{url} – Address of the service for viewing receipts.**

https://cheques-lk.orangedata.ru/

**{inn} – INN of the organization for which the receipt is punched**

**{document\_id} –** **ID of the document that was specified when it was created**

Example: <https://cheques-lk.orangedata.ru/9715225506/766eecfb05054fbab35d8c3d9f7d9d16>



# 5.Change Log

**Version 0.1.0 from 13.02.2017**

First version.

**Version 0.2.0 from 17.02.2017**

The contents of the check and the data issued upon request of the document status have been changed.

**Version 0.3.0 of 21.02.2017**

The time given as a result of breaking the check to the local one, clarification on the price of the goods, the sign of sending to the OFD and the receipt data has been changed.

**Version 0.4.0 from 23.03.2017**

The values ​​of the parameters of the taxation system and type of payment are added, the group attribute is added to the registration request.

**Version 0.5.0 from 13.04.2017**

The sign of sending to the OFD and the receipt data has been removed.

**Version 2.0.0 from 05.24.2017**

Added INN to the document status request.

Changed API version to v2.

**Version 2.1.0 from 29.05.2017**

Added clarification about the default group.

Added mention of swagger.

**Version 2.2.0 from 01.06.2017**

Added examples of signature generation on php and .net core and clarification on the used padding.

**Version 2.3.0 from 07.06.2017**

Added description of 401 errors.

Added an example of generating a signature on .net full.

**Version 2.4.0 from 14.06.2017**

Added an example of generating a signature in php.

**Version 2.5.0 from 19.06.2017**

The message format is aligned with what the API returns.

Added example response with an error.

**Version 2.6.0 from 20.06.2017**

Added an example of creating a signature on php using openssl.

**Version 2.7.0 from 26.06.2017**

Signature message example changed.

**Version 2.7.1 from 28.06.2017**

Fixed response code for an undetected document when checking status in accordance with the behavior of the system.

**Version 2.8.0 from 31.07.2017**

Added information about 503 errors when sending a document.

Added the position parameters of the document “Characteristic of the subject of calculation” and “Characteristic of the method of calculation”.

The maximum position, payment amounts have been clarified.

Increased number accuracy from 3 to 6 characters.

Where possible, the terminology is replaced with that used in the FFD.

**Version 2.9.0 from 03.08.2017**

Added information about the FFD version.

The moments that are worth paying attention to during development are described.

**Version 2.9.1 from 08.08.2017**

Clarifications have been made on the type of payment, for less confusion.

**Version 2.10.0 from 10.08.2017**

Added the ability to transfer agent data.

Added tag numbers corresponding to check data.

Added clarification about the FLC data and a link to the description of the FFD.

**Version 2.12.0 from 30.08.2017**

Added the ability to specify the key by which the signature was made.

**Version 2.13.0 from 11.10.2017**

Added the ability to transfer the tag 1084, an additional user attribute.

**Version 2.14.0 from 09.11.2017**

Added the ability to transfer the tag 1162, the commodity nomenclature code for the subject of calculation.

**Version 2.14.1 from 21.11.2017**

A clarification has been added on the restriction on the maximum price per unit of calculation item and the maximum number of calculation item.

**Version 2.14.2 from 12.12.2017**

The maximum length of the document ID has been increased from 32 to 64 characters.

**Version 2.14.3 from 17.12.2017**

Clarification was added to limit the maximum number of calculation items.

**Version 2.14.4 from 13.02.2018**

Fixed example of signature warped by word.

**Version 2.15.0 from 27.02.2018**

Added the ability to transfer tags 1225, Name of supplier, 1226, TIN of the supplier, 1171, Telephone number of the supplier associated with the subject of calculation.

**Version 2.16.0 from 15.05.2018**

Added the ability to transfer tags 1036, Machine number and 1009, Settlement address, 1187, Place of settlement.

Clarifications regarding length 1084, Additional user attribute.

Refinements regarding the maximum length of some parameters.

**Version 2.16.1 from 24.05.2018**

The restriction on the maximum number of positions is described.

**Version 2.17.0 from 19.06.2018**

Added tag transfer agent attribute for calculation item 1222, agent data 1223, unit of measurement for calculation item 1197, additional attribute for calculation item 1191.

**Version 2.18.0 from 01.08.2018**

Added tag transfer additional check requisite, 1192.

Changed the signature example to a more correct one with normal habits.

**Version 2.19.0 from 03.08.2018**

Added transfer of check tags from a new order of the Federal Tax Service:

* Buyer (client), 1227
* INN of the buyer (client), 1228
* Country of origin code, 1230
* Customs declaration number, 1231
* Excise tax, 1229

**Version 2.20.0 from 31.08.2018**

Added the ability to transfer tags 1036 Machine number, 1009 Settlement address, 1187 Settlement place in the correction receipt.

**Version 2.20.1 from 23.10.2018**

Additions and clarifications to the protocol.

**Version 2.21.0 from 01.12.2018**

Descriptions for tags 1139, 1141, 1151, 1153, 1162 have been changed.

**Version 2.22.0 from 28.04.2019**

Added values ​​for tag 1212 and 1030.

**Version 2.22.1 from 11.06.2019**

Added link to check.

**Version 2.23.0 from 15.06.2019**

The callbackUrl parameter has been added to requests for creating a check and correction check.

**Version 2.24.0 from 10.10.2019**

Changes were made to the code size of the product nomenclature, 1162.

**Version 2.24.1 of 25.11.2019**

The IP address of the test environment has been changed.

**Version 2.25.0 from 26.11.2019**

Possibility to transfer tags 1021 Cashier, 1117 Email address of the sender of the check, 1203 cashier's INN is added to the check.

**Version 2.26.0 from 09.12.2019**

Added a method for requesting the status of KKT in the group of clause 2.5.

Added the ability to transfer taxes for a check and subject of calculation, tags 1102, 1103, 1104, 1105, 1106, 1107, 1200.

**Version 2.27.0 from 27.02.2020**

Added new values for the attribute of the settlement item, tag 1212

The description of the correction of tag 1177 has been removed in accordance with the amendments to the FFD.

**Version 2.27.1 from 03.08.2020**

Added a note about fields **vat1Sum, vat2Sum, vat3Sum, vat4Sum, vat5Sum, vat6Sum** and tax rate numeration in the field **tax** (tag 1199).

Added a note about rounding in the field **totalSum** (tag 1020).

Updated possible values for field **paymentSubjectType** (tag 1212). Since version 2.27.0 values from 1 to 26 are allowed.

**Version 2.27.2 from 16.12.2020**

Added detailed information about a table of data Connection settings for production environment and two test environments.

Added detailed information about files for test in Git.

Added a note about Postman and SignatureGenerator.exe application.

Updated information about receipt size limitation. 32768 bytes was related to binary fiscal document size. JSON document can contain about 250 positions or 100 Kbytes.

**Version 2.28.0 from 07.20.2021**

The ignoreItemCodeCheck flag has been added to the body of the document creation request.

FFD 1.2 tags have been added to the request for document creation.

Added requests for generating a check for FFD 1.2 correction in clause 2.7 and clause 2.8.

Added requests for checking KM p.2.9 and p.2.10