Protocol for interaction with the Internet cash register

(OrangeData cloud service)

Version 2.30.0 from 05.12.2024

3

1. General description

1.1. Main information

Orange data is a cloud service that allows you to programmatically generate fiscal receipts. The service is used primarily for online stores, transport solutions and online commerce to generate receipts in accordance with Federal Law-54 dated May 22, 2003 "On the use of cash registers when making settlements in the Russian Federation."

To connect the cloud service, you need to register your organization on the website www . orangedata . ru , through the Personal Account, order the connection of physical fiscal devices that provide the formation of fiscal receipts. Next, you can generate fiscal checks using the API (described below). The service generates checks only in electronic form (without a printed copy).

Cash register equipment (KKT or CASS) is a special device that meets the requirements of the Federal Law of May 22 , 2003 N 54-FZ (dated July 3, 2018) "On the use of cash registers when making settlements in the Russian Federation". According to the law, all devices that can be used to generate fiscal checks are listed in a special register of cash registers. There are a number of different CCPs (with user interface, RS-232 or USB hardware interface, etc.). KKT (cash desk) can generate checks in paper form or in electronic form. The main functions common to all CCPs are:

- The cash register must correctly interact with the fiscal drive installed on this cash register (see description below).
- The check must contain **the organization's TIN** , as well as the required fields that are automatically generated by the fiscal accumulator:
  - serial number of the fiscal drive **(fsNumber),**
  - device serial number **(deviceSN) ,**
  - device registration number **(deviceRN),**
  - shift number (**shiftNumber**),
  - document number **(documentNumber),**
  - fiscal attribute **(fp)** .

  All of these fields are described below.

Orange cloud service data uses "NETPAY" cash registers (see Figure 1). Detailed characteristics of the "NETPAY" cash desk are published at http://paykiosk.ru/netpay . All cash registers are mounted in a server rack. The 2U high cabinet holds 196 CCPs and customers can "reserve" one or more units for their company. Each cash desk can be registered with the Federal Tax Service for only one company. Thus, the cash desk can be divided between several departments of the same company, but not between several customers.

Picture 1.

NETPAY checkout *with an inserted fiscal drive.*

Devices can be rack mounted.

The factory number (serial number) of the cash register (field SN) is the CCP number assigned by the manufacturer during production. According to the law, it consists of 16 digits. At the time of production, the manufacturer transmits to the Federal Tax Service information that the cash register with a certain serial number was produced and released for sale.

The registration number (field RN) is the number that the Federal Tax Service assigns to the cash register when the cash register user registers the cash register with the Federal Tax Service. The registration number, like the serial number, consists of 16 digits.

The registration number and serial number of the device are different values that exist in parallel. **The serial number** never changes. **The registration number** may change if the company cancels the registration of the cash register with the tax office and (after a pause) registers the cash desk again, or if the cash desk is "resold" to another organization and re-registered.

The fiscal drive (FN) is a special data storage device with a Java machine inside and hardware security. FN is installed in the cash register and controls the key functionality of the cash register. The fiscal drive provides the ability to record fiscal documents without the possibility of subsequent modification. Thus, no one can change a perfect transaction. Any correction is stored in the fiscal accumulator as a new transaction.

Each FN has a unique number containing 16 digits.

Figure 2 .

FN. The size of the device is 30 mm.

FN number (fsNumber) 16-digit FN serial number assigned by the manufacturer. Numbers of fiscal drives, numbers starting with four digits "9" (for example, 9999078900006784), are reserved for test FN. All other prefixes are used for the production environment.

TIN (taxpayer identification number) is a unique identifier that the Federal Tax Service assigns to each taxpayer. It contains 10 digits for companies and 12 digits for individuals and sole traders. For example, TIN 7725327863 is the identifier of the company ORANGE DATA JSC.

Shift number - when the cashier starts working, he opens the shift. FN generates a special transaction and reports it to the tax office. At the end of the working day, the cashier closes the shift and this message is sent to the tax office. A new FN starts with shift #1, the next #2 and so on.

Receipt number for the shift - the serial number of the cash receipt, SRF, correction cash receipt and SRF of the correction for the shift. When the cashier opens a shift, he generates document #1, and the next document #2, and so on...

Document number - a unique document number that FN assigns to any document. The new FN generates the first document #1, the next document #2 and so on...

Fiscal attribute (field fp) is a "checksum" that is generated by the fiscal accumulator based on: date, amount, TIN, FN number **,** document number, CRE registration number and CRE serial number. The Federal Tax Service uses a fiscal attribute to verify the authenticity of a check. If during the transfer of the check to the fiscal data operator (FDO) a discrepancy between the fiscal attribute and the data in the check is revealed, the check will not be accepted and the cash desk will be blocked after 30 days.

FFD (fiscal data format) is a binary document format that the cash desk generates and sends to the tax office. For the convenience of developers, the Oranjdata service uses the JSON format , however, inside the cash register, when generating a check, messages are converted to a binary format. At the moment, the fiscal data formats of the **FFD 1.05** and **FFD 1.2 versions are relevant.** Service " Orangedata " uses **FFD 1.05** . and **FFD 1.2.**

OFD (Fiscal data operator) is an organization responsible for receiving fiscal documents in binary format and sending information to the Federal Tax Service. There are about 15 authorized OFDs in Russia.

1.2. Connection to Orange data

OrangeData has two test environments and one production environment.

1.2.1. Two test environments

The differences between Test Environment 1 and Test Environment 2 are described below.

Test environment 1.	Test environment 2.
This test environment provides API functionality without a web cabinet. Device emulators are used instead of physical devices. Any tester can use the OrangeData 1 test environment for free. Test environment 1 can be used according to the INN of the test organization without a long configuration period*. Test Environment 1 is free and recommended for customer testing.	This test environment provides the functionality of both the web office and the API. Test Environment 2 uses physical devices equipped with Test Fiscal Memory (TSM). To set up test environment 2, it takes 3-5 days for each company to set up and install the device. This is not a free service. Users can use the TIN of the general test organization*. The test environment is mainly used by OrangeData developers and is not recommended for customers (some features may not be available when the OrangeData team is testing new versions).

* *Note:*

Test environment 1. Attribute «*inn*» «7727401209», attribute «*key*» «7727401209»;
Test environment 2. Attribute «*inn*» «7727401209», attribute «*key*» It is necessary to request by mail api@orangedata.ru. Staff will provide access.

Test environment connection settings:

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 office	Not available	https://test.orangedata.ru/lk/
Browser Accessibility Check URL	https://apip.orangedata.ru:2443/api/v2/	https://apip.orangedata.ru:12001/api/v2/
Swagger	https://apip.orangedata.ru:2443/swagger	https://apip.orangedata.ru:12001/swagger
S wagger JSON specification	https://apip.orangedata.ru:2443/swagger/v2/swagger.json	https://apip.orangedata.ru:12001/swagger/v2/swagger.json

1.2.2. Work environment

Parameter	Work environment
Base URL	https://api.orangedata.ru:12003
IP address	81.27.253.4 и 188.170.11.161
Web office	https://lk.orangedata.ru
Browser Accessibility Check URL	It is highly recommended to use the domain name for production requests. Add /api/v2/ to URL bar to check availability
swagger- ui	https://api.orangedata.ru:12003/swagger
Sw agger JSON specification	https://apip.orangedata.ru:12003/swagger/v2/swagger.json

1.2.3. Security

Security is provided in two steps:

1) Each client generates its own SSL certificate. This SSL certificate provides a secure connection between the client application and the OrangeData server . In a production environment, an SSL certificate is generated in the Client's Personal Account at https://lk.orangedata.ru The two test environments use the default test certificate. You can download the test certificate from Git at https://github.com/orangedata-official/API/tree/master/files_for_test/client_ca.crt .

   Note: in test environment 2, you can generate an SSL certificate in your personal account, however, in the test environment, the SSL certificates uploaded in your Personal Account are not used, only the certificate that is published in the GIT is used.

2) Any request to OrangeData to send any data must be signed. For example, to create documents, you need a signature. At the same time, the **"get check status"** request (GET request) does not need a signature. You can read more detailed information about document signing in sections **1.2.5 and 1.3** .

1.2.4. SSL certificate and keys for test environment

Both test environments (1 and 2) use an https connection with certificates. All certificates are published to Git at https://github.com/orangedata-official/API/tree/master/files_for_test

Description of files in the file_for_test directory in GIT

File	Description
client_ca.crt	Orangedata root certificate (with public key inside). All certificates of all clients are signed with the private key corresponding to this certificate. If the client certificate was signed by any other issuer, it will be rejected by the server.
client . key	Client private key for SSL connection for test environment.
client.crt	The client certificate for the test environment (signed with the private **key corresponding to the client_ca.crt** root certificate).
client.pfx	PFX client certificate for the test environment. PFX files contain both a public key and a private key. Those. such a file corresponds to two files at the same time - **client . key** and **client . crt** and can be obtained from these two files using the openssl utility . in the **client** file **. pfx** uses the password 1234. Depending on the software you are using, you may need either a **client . pfx** , or you can use **client . key** and **client . crt .**
private_key.xml	The user's private key for signing requests. More details are described below *in section No1.2.5.*
readme_eng.txt	Explanation in Russian .
readme_eng.txt	Explanation in English .

Description of the file in the GIT SignatureGenerator directory

File	Description
SignatureGenerator.exe.config	The configuration file for the SignatureGenerator program .
SignatureGenerator.exe	The executable file for the SignatureGenerator program .
SignatureGenerator.pdb	Data file .

1.2.5. Request Body Signing

Some requests (for example, all requests for the creation of fiscal documents) require signing of the request body.

To protect client messages from changes, a SHA256-RSA signature with a 2048-bit key is used. The signature is passed in the http header in the HEADER parameter **X - Signature** encoded using base64 encoding.

For two test environments, the signing key is used from the **private_key.xml file** , which is published in GIT at https://github.com/orangedata-official/API/tree/master/files_for_test

1.2.6. Message format and text encoding

Interaction with the cloud cash desk is carried out using web services. The HTTPS server uses client and server certificates for mutual authentication.

The message format is JSON. JSON size has no limit, but FN has a hardware limit of 32768 bytes for each financial document in binary form. This document size limit provides for about 250 items per receipt. FN will reject your check if you exceed the binary limit of 32768 bytes. We recommend using less than 250 items per receipt.

Cash desks create documents in accordance with **FFD 1.05** . and **FFD 1.2.** A detailed description of the format of fiscal documents can be found on the website of the Federal Tax Service at https://www.nalog.ru/html/sites/www.new.nalog.ru/doc/pril2_fns229_210317.docx

OrangeData API uses **UTF-8 encoding** , but all types of fiscal drives only support **CP866 encoding.** Due to this design feature of fiscal drives, the service converts data from your JSON request with **UTF-8 encoding** into a binary data packet of the Tag -Length -Value (TLV) type with **CP866 encoding when punching a check.** Since **CP866 encoding has** fewer characters than **UTF-8** , all "unsupported" characters for **CP 866** are ignored. If the system receives an "unsupported" character in **CP866**

, then it will not be included in the check (for example, in **CP866 encoding** there are no angle quotes '« ' and '"', so a string with angle brackets, for example, 'Juice "Dream"' will be replaced by string without angle brackets 'Dream Juice').

CCT performs "moderate" format-logical control of data, i.e. Format violations that the fiscal accumulator and OFD "prohibit" are controlled and prohibited by the service (an error is returned). However, violations of the FFD, which "miss" both FN and OFD and the Federal Tax Service - the Orange Data service also "allows". For example, according to the FFD, tag 1008 (client's phone number or client's e-mail) has the format +{N} or {X}@{X}, but the FTS and OFD allow you to send any string from 1 to 64 characters long. It makes no sense to carry out stricter control than the Federal Tax Service. However, we recommend that you comply with the requirements of the formats in full. It is recommended to familiarize yourself with the FFD formats on the website of the Federal Tax Service at https://www.nalog.ru/html/sites/www.new.nalog.ru/doc/pril2_fns229_210317.docx

Please take care of proper handling of network connection errors if an error occurred while sending a fiscal document, send it several times with a timeout. We recommend 5 attempts with an interval of 10-20 seconds (numbers are given as an example and may vary depending on the specific situation).

1.3. Signature example request

```
POST / api / v 2 / documents / HTTP /1.0

Connection: close
Content-Type: application/ json ; charset=utf-8
Accept: */*
Host: appip.orangedata.ru
Content Length: 509
X-Signature:  Ql1ZoZfYQo9NAgJQXUsBKQHQjqkICmn9b2jCfUyUDQYExMRtfhciGEs35dUFisVWiveeWF9dgBkzbJtNgEyQoNR  XE  +
37ZMmEjS / SizRVpeAIKfpAx8GseQnv3ssw2NtvAELMduAgsz3hC8mIrOt // CPmQABd8X21NqfzZY4zVZgOTu / q2fNXBlv + Uyv
J5amwzMkh7aX5g0RQDBAQ7UovscpQY41t43rdYZ6UVVv  /  OHEfExs7vEZmi7c3tdUJVf98Bli3oh5RfMnsb1  /  m2kZBqhxC0zThCKd
yGlmFNM + tAlVqbKHDkulZRgwarwXbL17y / xbPeLlQZLzIU2esYTMulg ==

{
" id ": "2_loc_z5bYWHvD",
" inn ": "1234567890",
" group ": "Main",
" key ": "1234567890",
" content ": {
" type ": 1,
" positions ": [
{
" quantity ": 2,
" price ": 10,
```

```
" tax ": 1,
" text ": " Pickup ru ",
" paymentMethodType ": 4,
paymentSubjectType : 1
},
{
" quantity ": 1,
" price ": 0,
" tax ": 1,
" text ": "Delivery",
" paymentMethodType ": 4,
paymentSubjectType : 4
}
],
" checkClose ": {
" payments ": [
{
" type ": 2,
" amount ": 20
}
],
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>
```

14

```
<Q>
1IRVid5SsDrOwJQAEKkdT436Xeb0sVWe9AcU8JyaCEEMj0NpzownNbIrebPofMYdDHikopQpr2XqxZYDbb7AneoHkhEV26TfpPVbN4wBJFXih3
lAP2n5hqhgqHGp5Wq2Lu7jUS376Ruw3bhwW + MiWpXv1xhMTZ8AtDfnZFFNvOM = < / Q>
<DP>
Fo5KiNCJCtCbpFfH4XVM5UJdXPXTbNBHBdlYMJ9AddTl5Ijrt50ExgLFu4oMPMsYXryS61LI2WT5XCqIvmbcnhYbambgWLOKYuZUUYSr2kS67S
o5FUCunWaGhTdx2bRLQVqwm6kiXDPDnMRAViiCHXWqk / 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       +
WiTEyfShNV9qVD0PBltz3omch1GjpFhQn6OaRvraeIDH9HXttb3Fojr2zzYG4yrrYbPSRWoYj63ZwiIP2O7zdl0caGQHezfNcYa2N0NTG99DGc
3 / q6EnhlvjWQsSbiEjmxcPx8fmV1i4DoflMQ383nsixAFapgrROUAtCgMvhWn1kSeoojKd + e4eKZxa / SNYulsBJWNFkmo1CZH4YtqlPM
+ IwYeDUOnOUGNxGurRZ3qQdWs2N2ZQhnrvlh + zpzurD2hwAz6gQXP7mxxMR1xHtAD8XQ + w4OiJK6VwjoIQ == </ D>
</ RSAKeyValue >
```

Signature generation in php with https :// github . com / phpseclib / phpseclib

```php
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));
}
```

An example of generating a signature in php using openssl . This method requires a key in PEM format. You can convert xml to PEM with this converter https://superdry.apphb.com/tools/online-rsa-key-converter:

```php
function computeSignature ($data)
```

```php
{
$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);
```

```php
        openssl_private_encrypt ( $data, $res, $ pk ) ;

        return base64_encode($res);
```

Message data in php :

```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 the standard class library . net core :

```csharp
 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));
    }
}
```

C# signatures using the standard library of the . net , complete:

```
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. API testing with Postman and Signature Generator .

You can test the OrangeData API with the Postman application . Request body signing can be done by the **SignatureGenerator application** , which is published on Git at https://github.com/orangedata-official/API/tree/master/SignatureGenerator

2. Requests

2.1 Creating a receipt

Checks are generated in 3 stages:

Stage 1:	The client application (online store) sends an asynchronous request to generate a fiscal document. The server stores the request data in a processing queue.
Stage 2:	The server extracts requests from the processing queue one by one and transfers them to the first free cash register for "punching" the fiscal document. After the cash desk returns the result of the work, the information is stored in the database.
Stage 3 :	The client application (online store) requests the status of a fiscal document and a set of fiscal parameters (fiscal number, fiscal attribute, etc.).

Each asynchronous request contains a "custom" document ID. This field must be unique within the organization. The identifier field is used to monitor the status of requests in step 3 (see section 2.2.).

Orange personal account data . The number of devices determines the number of checks that a client can generate every second. The maximum productivity of one cash register is one check per second. If a company needs more checks per second, more cash registers are needed.

All cash desks of each organization are combined into groups. Each checkout can be linked to one group. Groups allow customers to indicate in requests that it is necessary to "break a check" at certain cash desks: if the initial request contains the name of a group, devices from the specified group will be used to generate a check. If the group is not specified or NULL, the default group will be used to generate the receipt.

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

Type of personal account	Default group name	Group name for FFD 1.2
Standard	Main	Main_2
vending	Vend	Vend_2
Transport	Transp or Courier*	Transp_2 or Courier_2

* The group will depend on what settings will be selected during the fiscalization of the cash register in the Orange Personal Account data .
Automatic devices – Transp Group
Manual payment terminals – Courier Group

You can check the names of groups that are relevant for your organization in your personal account, you can also create and rename groups.

The addition to the name "_2" will indicate that the group works only with cash registers fiscalized for work with FFD 1.2

If you have groups with a different name, you will need to contact the specified mail (api@orangedata.ru or help@orangedata.ru) to change the name of the groups.

By default, the system provides a uniform load on the cash registers and selects the "first free" matching device (with the correct TIN and the corresponding group) to generate receipts. This logic can be changed in the settings.

Service Orange Data stores check generation requests for no more than 30 days. Long-term " *storage of requests in a queue* " may occur, for example, if the client specified a group of cash desks in the request parameters (the **group parameter**), but no cash register is "linked" to this group in the personal account, or if the client has only one cash desk and it is " got up" due to blocking by the fiscal drive or for other reasons. If a request has been " *in the queue* " for more than 30 calendar days, it will not be fulfilled.

After the *check* has been punched, the result of the transaction is stored in the database for another 3 months. After 3 months, the data from the database is moved to the archive.

Inquiry: **POST / api / v2 / documents /**

2.1.1 Request body

id	Document ID	String from 1 to 64 characters.	The parameter is required.
inn	TIN of the organization for which the check is made	A string of 10 or 12 characters.	The parameter is required.
group	Group of devices with which the check will be punched	String from 1 to 32 characters or null	The parameter is required
content	Document content	Structure of clause 2.1.1.1	
key	The name of the key to be used to verify the signature. For clients, their TIN is used, for	String from 1 to 32 characters.	The parameter is required.

	partners and paying agents, the code with a mask is 301 ****, for vending 401 ****.		
callbackUrl	URL for sending check processing results by POST request	String from 1 to 1024 characters or null	The parameter is optional
callbackApiKey	API key for calling the callback	String from 1 to 3072 characters or null	The parameter is optional
meta	Request metadata	String from 1 to 128 characters or null	The parameter is optional
ignoreItemCodeCheck	Flag indicating whether the CM check should be ignored. If the flag is not specified, then to generate a check, all CMs must successfully pass the check: 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 status 422 Unprocessable entity	Boolean value The default value is false .	optional field.

2.1.1.1 Document content

Attribute	Description	A comment	FFD 1.05	FFD 1.2	obligatory
ffdVersion	FFD version number, 1209 2 - FFD 1.05, default value 4 - FFD 1.2	Number.	Yes	Yes	Optional for FFD 1.05 Mandatory for FFD 1.2
type	Settlement sign, 1054: 1. Coming 2. return parish 3. Consumption 4. Expense return	Number.	Yes	Yes	FFD 1.05 - yes FFD 1.2 - yes

positions	List of calculation items, 1059	Array of structures p.2.1.1.2	Yes	Yes	
checkClose	Receipt closing options	Structure of clause 2.1.1.3	Yes	Yes	
with customerContact	Phone or email address of the buyer, 1008	String from 1 to 64 characters, format +{C} or {C}@{C}, {none}.	Yes	Yes	FFD 1.05 - yes FFD 1.2 - yes
agentType	Sign of an agent, 1057 . **A bit** field where the bit number indicates that the user providing the service to the buyer (client) is: 0 (1) – bank paying agent 1 (2) – bank payment subagent 2 (4) - paying agent 3 (8) – payment subagent 4 (16) - attorney 5 (32) - commission agent 6 (64) - other agent *A cash receipt (SRF) may contain the "agent attribute" (tag 1057) requisites only if the registration report and (or) the current re-registration report contains the "agent attribute" variable (tag 1057) that has a value identical to the value of the "agent attribute" variable » (tag 1057) cash receipt.*	Number from 1 to 127 , optional field	Yes	No	FFD 1.05 - yes FFD 1.2 - yes

applicationTransferOperatorPhoneNumbers	Phone number of transfer operator, 1075	Array of strings from 1 to 19 characters long, format +{C}.	Yes	No	FFD 1.05 - No FFD 1.2 - No
applicationAgentOperation	Operation payment agent, 1044	A string with a length of 1 to 24 characters.	Yes	No	FFD 1.05 - No FFD 1.2 - No
paymentAgentPhoneNumbers	Payment agent phone number, 1073	Array of strings from 1 to 19 characters long, format +{C}.	Yes	No	FFD 1.05 - No FFD 1.2 - No
paymentOperatorPhoneNumbers	Phone number of payment acceptance operator, 1074	Array of strings from 1 to 19 characters long, format +{C}.	Yes	No	FFD 1.05 - No FFD 1.2 - No
paymentOperatorName	Name of transfer operator, 1026	String length from 1 to 64 characters	Yes	No	FFD 1.05 - No FFD 1.2 - No
paymentOperatorAddress	Transfer operator address, 1005	A string from 1 to 243 characters long.	Yes	No	FFD 1.05 - No FFD 1.2 - No
paymentOperatorINN	TIN of the transfer operator, 1016	A string of 10 to 12 characters, format DDDDDDDDDD .	Yes	No	FFD 1.05 - No FFD 1.2 - No
supplierPhoneNumbers	Supplier phone number, 1171	Array of strings from 1 to 19 characters long, format +{C}.	Yes	No	FFD 1.05 - No FFD 1.2 - No
additionalUserAttribute	Additional username, 1084	The structure of clause 2.1.1. five	Yes	Yes	FFD 1.05 - No FFD 1.2 - No
additionalAttribute	Additional details of the check (BSO) , 119 2	String from 1 to 16 characters.	Yes	Yes	FFD 1.05 - No FFD 1.2 - No
automatNumber	Machine number, 1036	A string with a length of 1 to 20 characters, a	Yes	Yes	FFD 1.05 – yes for vending and transport.

		mandatory field if the group has the attribute of transferring machine data, otherwise it should not be transferred. **Only for vending and transport.**			FFD 1.2 – yes for vending and transport.
settlementAddress	Settlement address, 1009	A string from 1 to 243 characters long, a mandatory field if the group has the automaton data transmission flag, otherwise it should not be transmitted . **For vending , transport and couriers.**	Yes	Yes	FFD 1.05 – yes for vending , transport, couriers. FFD 1.2 – yes for vending , transport, couriers.
settlementPlace	Place of settlement, 1187	A string from 1 to 243 characters long, a mandatory field if the group has the automaton data transmission flag, otherwise it should not be transmitted. **For vending ,**	Yes	Yes	FFD 1.05 – yes for vending , transport, couriers. FFD 1.2 – yes for vending , transport, couriers.

		transport and couriers.			
customer	Buyer (client), 1227	String from 1 to 243 characters.	Yes	No	FFD 1.05 - No FFD 1.2 - No
customerINN	TIN of the buyer (client), 12 28	A string of 10 to 12 characters, format DDDDDDDDDD.	Yes	No	FFD 1.05 - No FFD 1.2 - No
customer info	Information about the buyer (client), 1256	The structure of clause 2.1.1.8.	Not	Yes	FFD 1.05 - No FFD 1.2 - No
cashier	Cashier, 1021	A string from 1 to 64 characters, a mandatory field if the group has the sign of passing courier parameters, otherwise it should not be sent. **Mandatory for couriers**	Yes	Yes	FFD 1.05 – yes for couriers. FFD 1.2 for couriers.
cashier INN	TIN of the cashier, 1203	A string of 12 characters, format DDDDDDDDDDDDD.	Yes	Yes	FFD 1.05 - No FFD 1.2 - No
senderEmail	Check sender's email address, 1117	String from 1 to 64 characters, format +{C} or {C}@{C}, {none}.	Yes	Yes	FFD 1.05 - No FFD 1.2 - No
totalSum	The amount of the calculation indicated in the check (BSO), 1020.	Decimal number accurate to 2 characters after the dot.	Yes	Yes	FFD 1.05 - No FFD 1.2 - No

	Fiscal Document Formats allow rounding down in this field. Literally, the FFD requirement sounds like this: "The value in rubles, excluding kopecks, must be equal to the value of the sum of all details "the cost of the subject of calculation, taking into account discounts and markups" (tag 1043) in rubles, excluding kopecks." To avoid inconsistencies, we recommend using the exact value without rounding.				
vat1Sum	The amount of the VAT check at the rate of 20%, 1102	Decimal number accurate to 2 characters after the dot.	Yes	Yes	FFD 1.05 - No FFD 1.2 - No
vat2Sum	The amount of VAT check at a rate of 10%, 1103	Decimal number accurate to 2 characters after the dot.	Yes	Yes	FFD 1.05 - No FFD 1.2 - No
vat3Sum	Settlement amount on a check with VAT at a rate of 0%, 1104	Decimal number accurate to 2 characters after the dot.	Yes	Yes	FFD 1.05 - No FFD 1.2 - No
vat4Sum	Amount of check settlement without VAT, 1105	Decimal number accurate to 2 characters after the dot.	Yes	Yes	FFD 1.05 - No FFD 1.2 - No

vat5Sum	The amount of the VAT check according to calc . rate 20/120, 1106	Decimal number accurate to 2 characters after the dot.	Yes	Yes	FFD 1.05 - No FFD 1.2 - No
vat6Sum	The amount of the VAT check according to calc . rate 10/110, 1107	Decimal number accurate to 2 characters after the dot.	Yes	Yes	FFD 1.05 - No FFD 1.2 - No
vat7Sum	The amount of the VAT check at the rate of **5%**	Decimal number accurate to 2 characters after the dot.	Yes	Yes	FFD 1.05 - No FFD 1.2 - No
vat8Sum	The amount of the VAT check at the rate of 7%	Decimal number accurate to 2 characters after the dot.	Yes	Yes	FFD 1.05 - No FFD 1.2 - No
vat9Sum	The amount of the VAT check according to calc . rate 5/105	Decimal number accurate to 2 characters after the dot.	Yes	Yes	FFD 1.05 - No FFD 1.2 - No
vat10Sum	The amount of the VAT check according to calc . rate 7/107	Decimal number accurate to 2 characters after the dot.	Yes	Yes	FFD 1.05 - No FFD 1.2 - No
operationalAttribute	Operating details of the check, 1270	Structure of clause 2.1.1.9	No	Yes	FFD 1.05 - No FFD 1.2 - No
industryAttribute	Branch requisite of the check, 1261	Structure of clause 2.1.1.10	No	Yes	FFD 1.05 - No FFD 1.2 - No

Note: for historical reasons, the numbering of tax rates **vat1Sum, vat2Sum, vat3Sum, vat4Sum, vat5Sum, vat6Sum, vat7Sum, vat8Sum, vat9Sum, vat10Sum** in OrangeData formats is slightly different from the numbering of tax rates in the FFD for tag 1199. When developing integration, one should strictly adhere to this manual.

For transport/ vending , one of the required parameters is tags (**1036, 1009, 1187**)
For couriers, one of the required parameters is tags (**1021, 1009, 1187**)

The **Customer** and **customerINN** parameters are changed to one **customerInfo parameter .**

2.1.1.2 Subject of calculation

Attribute	Description	A comment	FFD 1.05	FFD 1.2	obligatory
quantity	Number of subject of calculation, 1023	Decimal number up to 6 characters after the dot *.	Yes	Yes	FFD 1.05 - yes FFD 1.2 - yes
price	Price per unit of the subject of calculation, taking into account discounts and markups, 1079	Decimal number up to 2 characters after the dot * .	Yes	Yes	FFD 1.05 - yes FFD 1.2 - yes
tax	VAT rate, 1199: 1 - VAT rate 20% 2 - VAT rate 10% 3 - VAT rate calculated. 20/120 4 - VAT rate calculated. 10/110 5 - VAT rate 0% 6 - VAT exempt 7 - VAT rate 5% 8 - VAT rate 7% 9 - VAT rate calculated. 5/105 10 - VAT rate calculated. 7/107	Number from 1 to 6.	Yes	Yes	FFD 1.05 - yes FFD 1.2 - yes
taxSum	The amount of VAT for the subject of calculation, 1200 *The parameter is relevant for VAT rates 1-4, 7-10.For rates 5 and 6, this tag is not transmitted in the subject of calculation, the transmitted value is ignored. If you pass the value 0, then the tag will not be written.*	Decimal number accurate to 2 characters after the dot.	Yes	Yes	ФФД 1.05 – No ФФД 1.2 - No
text	Name of the subject of calculation, 1030	String up to 128 characters .	Yes	Yes	FFD 1.05 - yes FFD 1.2 - yes

paymentMethodType	Settlement method attribute, 1214: 1 – Prepayment 100% 2 - Partial prepayment 3 - Advance 4 - Full settlement 5 - Partial settlement and credit 6 - Transfer on credit 7 - payment of the loan	A number between 1 and 7 , or null . If null is passed, then the value 4, Full settlement, will be passed to the subject of calculation .	Yes	Yes	FFD 1.05 - yes FFD 1.2 - yes
paymentSubjectType	Settlement subject attribute, 1212: 1 - Product 2 - Excisable goods 3 - Work 4 - Service 5 – Gambling bet 6 - Winning a game of chance 7 - Lottery ticket 8 - Winning the lottery 9 - Provision of RIA 10 - Payment 11 - Agency fee 12 - Payout 13 - Other subject of calculation 14 - Property law 15 – To non-operating income 16 - Other payments and contributions 17 – Sales tax 18 - Resort fee 19 - Pledge 20 - Consumption 21 - Contributions for compulsory pension insurance for individual entrepreneurs 22 - Contributions for compulsory pension insurance	A number from 1 to 33, or null . If null is passed , then the value 1, Goods, will be passed to the subject of calculation. For values 15 and 16 see note.	Yes	Yes	FFD 1.05 - yes FFD 1.2 - yes

	23 - Contributions for compulsory health insurance for individual entrepreneurs 24 – Compulsory health insurance contributions 25 – Compulsory social insurance contributions 26 - Casino payment 27 - Issuance of funds 30 - ATNM (not having a marking code) 31 - ATM (having a marking code) 32 - TNM 33 - TM				
nomenclature code	Commodity item code, 1162	A string containing a **base 64** encoded array of 8 to 32 bytes, or null .	Yes	No	FFD 1.05 - No FFD 1.2 - No
itemCode	Marking code, 2000 The KM itself, as it was read by the scanner. After verification, it will be added to 1163 as one of the tags "KT GS1.M" (tag 1305), or "KT KMK" (tag 1306), or "unrecognized product code" (tag 1300).	String from 1 to 223 characters.	No	Yes	FFD 1.05 - No FFD 1.2 - No
supplierInfo	Vendor data , 1224	Structure p.2.1.1.6, or null	Yes	Yes	
supplierINN	Supplier TIN , 1226	A string of 10 to 12 characters, format DDDDDDDDDDDD	Yes	Yes	FFD 1.05 - No FFD 1.2 - No
agentType	Sign of the agent on the subject of calculation, 1222 . **A bit** field where the bit number indicates	Number from 1 to 127	Yes	Yes	FFD 1.05 - No FFD 1.2 - No

	that the user providing the service to the buyer (client) is: 0 (1) – bank paying agent 1 (2) – bank payment subagent 2 (4) - paying agent 3 (8) – payment subagent 4 (16) - attorney 5 (32) - commission agent 6 (64) - other agent				
agentInfo	Agent data, 1223	Structure p.2.1.1.7, or null . Currently, serialized tag data must be limited to 243 bytes.	Yes	Yes	FFD 1.05 - No FFD 1.2 - No
unitOfMeasurement	Unit of measurement of the subject of calculation, 1197	String from 1 to 16 characters	Yes	No	FFD 1.05 - No FFD 1.2 - No
quantityMeasurementUnit	Measure of the quantity of the subject of calculation, 2108	A number from 0 to 255.	No	Yes	FFD 1.05 - No FFD 1.2 - No
additionalAttribute	Additional details of the subject of calculation, 1191	String from 1 to 64 characters.	Yes	Yes	FFD 1.05 - No FFD 1.2 - No
manufacturerCountryCode	Code of the country of origin of goods, 1230	A string with a length of 1 to 3 characters, CCC format. The service will automatically pad the string up to 3 characters with spaces.	Yes	Yes	FFD 1.05 - No FFD 1.2 - No
customsDeclarationNumber	Customs declaration number, 1231	String from 1 to 32 characters.	Yes	Yes	FFD 1.05 - No FFD 1.2 - No
excise	Excise, 1229	Decimal number accurate to 2 characters after the dot.	Yes	Yes	FFD 1.05 - No FFD 1.2 - No
unitTaxSum	The amount of VAT per unit of the subject of calculation, 1198	Decimal number accurate to 2	Yes	Yes	FFD 1.05 - No FFD 1.2 - No

31

		characters after the dot.			
fractionalQuantity	Fractional quantity of marked goods, 1291	The structure of clause 2.1.1.12.	No	Yes	FFD 1.05 - No FFD 1.2 - No
industryAttribute	Industry requisite of the subject of calculation, 1260	The structure of clause 2.1.1.10.	No	Yes	FFD 1.05 - No FFD 1.2 - No
barcodes	Barcodes of the subject of calculation	The structure of clause 2.1.1.11.	No	Yes	FFD 1.05 - No FFD 1.2 - No

The **nomenclatureCode parameter** is used for **FFD 1.05** , **itemCode** is used for **FFD 1.2.**

The **unitOfMeasurement parameter is** used for **FFD 1.05** , **quantityMeasurementUnit** is used for **FFD 1.2**

2.1.1.2.1 Name of the subject of calculation

Props value	The attribute "name of the subject of calculation" (tag 1030) contains information	PF format
"1"	income from equity participation in other organizations, with the exception of income directed to pay for additional shares (stakes) placed among shareholders (members) of the organization	income from equity participation in other organizations
"2"	income in the form of a positive (negative) exchange rate difference resulting from the deviation of the sale (purchase) rate of foreign currency from the official rate established by the Central Bank of the Russian Federation on the date of transfer of ownership of foreign currency (the specifics of determining banks' income from these operations are established by Article 290 of the Tax Code Russian Federation)	income in the form of a foreign exchange difference resulting from the deviation of the selling (purchasing) rate of foreign currency from the official rate
"3"	income in the form of fines, penalties and (or) other	income in the form of fines,

	sanctions recognized by the debtor or payable by the debtor on the basis of a court decision that has entered into force, for violation of contractual obligations, as well as amounts of compensation for losses or damage	penalties and (or) other sanctions payable by the debtor for breach of contractual obligations
"4"	income from the lease of property (including land plots) (sublease), 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 lease of property (including land plots) (sublease)
"5"	income from the granting for use of rights to the results of intellectual activity and rights to equivalent means of individualization (in particular, from granting for use of rights arising from patents for inventions, utility models, industrial designs), if such income is not determined by the taxpayer in the manner established 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, credit, bank account, bank deposit agreements, as well as on securities and other debt obligations (the specifics for determining bank income in the form of interest are established by Article 290 of the Tax Code of the Russian Federation)	income in the form of 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 expenses in the manner and on the conditions established by 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 in the form of recoverable 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 in the form of 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 the manner prescribed by Article 278 of the Tax Code of the Russian Federation	income in the form of income distributed in favor of the taxpayer with his participation in a simple partnership
"10"	income in the form of income of previous years, revealed in the reporting (tax) period	income in the form of income of previous years, revealed in the reporting (tax) period
"11"	income in the form of a positive foreign exchange difference, except for a positive foreign exchange difference arising from the revaluation of issued (received) advances	foreign exchange gain
"12"	income in the form of fixed assets and intangible assets received free of charge in accordance with international treaties of the Russian Federation or with the legislation of the Russian Federation by nuclear power plants to improve their safety, used not for production purposes	income in the form of fixed assets and intangible assets received free of charge by nuclear power plants
"13"	income in the form of the value of materials or other property received during dismantling or dismantling during the liquidation of fixed assets being decommissioned (except as provided for by 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 upon liquidation of decommissioned fixed assets
"14"	income in the form of property (including funds) used for other purposes, works, services received as part of charitable activities (including in the form of charitable assistance, donations), earmarked receipts, earmarked financing, with the exception of budgetary funds. With regard to budgetary funds used for other than their intended purpose, the norms of the budgetary legislation of the Russian Federation are applied.	income in the form of property, works, services used for other purposes
"15"	income in the form of funds used not for their intended	income in the form of funds not

	purpose by enterprises and organizations that include especially radiation hazardous and nuclear hazardous industries and facilities, funds intended for the formation of reserves to ensure the safety of these industries and facilities at all stages of their life cycle and development in accordance with legislation of the Russian Federation on the use of atomic energy	used for their intended purpose, intended for the formation of reserves to ensure the safety of production
"16"	income in the form of amounts by which in the reporting (tax) period there was a decrease in the authorized (share) capital (fund) of the organization, if such a decrease was carried out with a simultaneous refusal to return the value of the corresponding part of the contributions (contributions) to shareholders (participants) of the organization (except for cases provided for by subparagraph 17 of paragraph 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 refunds from a non-profit organization of previously paid contributions (contributions) in the event that such contributions (contributions) were previously taken into account as expenses when forming the tax base	income in the form of refunds from a non-profit organization of previously paid contributions (contributions)
"18"	income in the form of amounts of accounts payable (obligations to creditors) written off due to the expiration of the limitation period or on other grounds, with the exception of cases provided for by subparagraph 21 of paragraph 1 of article 251 of the Tax Code of the Russian Federation. The provisions of this paragraph do not apply to the write-off of accounts payable by a mortgage agent in the form of obligations to the owners of mortgage-backed bonds, as well as to the write-off of accounts payable by a specialized company in the form of obligations to the owners of bonds issued by it.	income in the form of amounts of accounts payable written off due to the expiration of the limitation period or on other grounds
"19"	income in the form of income received from operations	income in the form of income

	with derivative financial instruments, taking into account the provisions of Articles 301 - 305 of the Tax Code of the Russian Federation	received from operations with derivative financial instruments
"20"	income in the form of the value of surplus inventories and other property that are identified as a result of the inventory	income in the form of the value of surplus inventories and other property that are identified as a result of the inventory
"21"	income in the form of the cost of media products and book products subject to replacement upon return or write-off of 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 assessment of the value specified in this paragraph of the products is carried out in accordance with the procedure for assessing 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, subject to replacement upon return or write-off
"22"	income in the form of adjustments to the taxpayer's profit due to the application of methods for determining, for tax purposes, the conformity 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 taxpayer profit adjustment amounts
"23"	income in the form of the monetary equivalent of real estate and (or) securities returned to the donor or his successors, transferred to replenish the target capital of a non-profit organization in the manner established by Federal Law No. organizations" (Sobraniye zakonodatelstva Rossiyskoy Federatsii, 2006, N 1 (part 1), art. 38; 2013, N 30 (part 1), art. 4084)	income in the form of a returned cash equivalent of real estate and (or) securities transferred to replenish the target capital of a non-profit organization
"24"	income in the form of the difference between the amount of tax deductions from the excise amounts accrued when performing the transactions specified in subparagraphs	income in the form of the difference between the amount of tax deductions from excise

	21, 23 - 33 of paragraph 1 of Article 182 of the Tax Code of the Russian Federation, and the indicated excise amounts	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 profits of a controlled foreign company
"26"	insurance premiums for compulsory pension insurance	contributions to the OPS
"27"	insurance premiums for compulsory social insurance in case of temporary disability and in connection with motherhood	contributions to the OSS due to disability
"28"	compulsory health insurance premiums	contributions to compulsory health insurance
"29"	insurance premiums for compulsory social insurance against accidents at work and occupational diseases	contributions to OSS from accidents
"30"	expenses for the payment in accordance with the legislation of the Russian Federation of temporary disability benefits (with the exception of accidents at work and occupational diseases) for the days of temporary disability of the employee, which are paid at the expense of the employer and the number of which is established by Federal Law of December 29, 2006 N 255- Federal Law "On compulsory social insurance in case of temporary disability and in connection with motherhood" (Sobraniye Zakonodatelstva Rossiyskoy Federatsii, 2007, N 1 (part 1), art. 18; 2018, N 11, art. 1591), in part, not covered by insurance payments made to employees by insurance organizations that have licenses issued in accordance with the legislation of the	temporary disability benefit

	Russian Federation to carry out the relevant type of activity, under contracts with employers in favor of employees in case of their temporary disability (with the exception of accidents at work and occupational diseases) for days of times 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 Case of Temporary Disability and in Connection with Motherhood"	
"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 to carry out the relevant type of activity, in favor of employees in case of their temporary disability (except for accidents at work and occupational diseases) for 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 Case of Temporary Disability and in Connection with Motherhood".	voluntary personal insurance payments

2.1.1.3 Check closing parameters

payments	Payments	Array of structures p.2.1.1.4	obligatory
taxationSystem	Taxation system, 1055: 0 - General, DOS 1 - Simplified income, STS income 2 - Simplified income minus expense, STS income - expense 3 - Single tax on imputed income, UTII 4 - Single agricultural tax, UST 5 - Patent Taxation System, Patent	Number from 0 to 5.	FFD 1.05 - Yes FFD 1.2 - Yes

Note:
Attention ! From 01/01/2021 UTII has been cancelled. To select a suitable tax regime, you can use a special <u>service</u> .

2.1.1. 4 Payment

type	Payment type: 1 - cash check amount, 1031 2 – cashless check amount, 1081 14 - the amount of the check in advance (setting off the advance and (or) previous payments), 1215 15 – postpaid check amount (on credit), 1216 16 - the amount of the check (BSO) with a counter provision, 1217	Number from 1 to 16.	FFD 1.05 - Yes FFD 1.2 - Yes
a mount	Payment amount	Decimal number accurate to 2 characters after the dot *. The parameter is required.	FFD 1.05 - Yes FFD 1.2 - Yes

* The maximum check amount 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 quantity of the settlement item is 281,474,976.710,655 .

2.1.1. 5 Additional user details

name	Name of additional user attribute, 1085	String from 1 to 64 characters. *name + value attributes must not exceed 235 characters in total.*	FFD 1.05 - No FFD 1.2 - No
value	The value of the additional user attribute, 1086	String from 1 to 234 characters. *name + value attributes must not exceed 235 characters in total.*	FFD 1.05 - No FFD 1.2 - No

2.1.1.6 Supplier details

phoneNumbers	Supplier phone number 1171	Array of strings from 1 to 19 characters long, format +{C}.	FFD 1.05 - No FFD 1.2 - No
name	Name of supplier , 1225	String up to 239 characters. *Attention: 243 characters data includes supplier's phones + 4 characters per phone.* For example, if 2 supplier's phone numbers are 12 and 14	FFD 1.05 - No FFD 1.2 - No

| | | characters long, then the maximum length of the supplier's name will be 239 - (12 + 4) - (14 + 4) = 205 characters | |

2.1.1.7 Agent data

applicationTransferOperatorPhoneNumbers	Phone number of transfer operator, 1075	Array of strings from 1 to 19 characters, format +{C }.	FFD 1.05 - No FFD 1.2 - No
applicationAgentOperation	Operation payment agent, 1044	A string with a length of 1 to 24 characters.	FFD 1.05 - No FFD 1.2 - No
paymentAgentPhoneNumbers	Payment agent phone number, 1073	Array of strings from 1 to 19 characters long, format +{C}.	FFD 1.05 - No FFD 1.2 - No
paymentOperatorPhoneNumbers	Phone number of payment acceptance operator, 1074	Array of strings from 1 to 19 characters long, format +{C}.	FFD 1.05 - No FFD 1.2 - No
paymentOperatorName	Name of transfer operator, 1026	A string from 1 to 64 characters long.	FFD 1.05 - No FFD 1.2 - No
paymentOperatorAddress	Transfer operator address, 1005	A string from 1 to 243 characters long.	FFD 1.05 - No FFD 1.2 - No
paymentOperatorINN	TIN of the transfer operator, 1016	A string of 10 to 12 characters, format DDDDDDDDDD.	FFD 1.05 - No FFD 1.2 - No

2.1.1. 8 Information about the buyer (client) , 1256

name	Buyer (client), 1227	String from 1 to 239 characters.	FFD 1.05 - No FFD 1.2 - No
inn	TIN of the buyer (client), 12 28	A string of 10 to 12 characters, format DDDDDDDDDD.	FFD 1.05 - No FFD 1.2 - No
birthDate	Date of birth of the buyer (client), 1243	String of 10 characters, format DD.MM.YYYY .	FFD 1.05 - No FFD 1.2 - No

citizenship	Citizenship, 1244	A string with a length of 1 to 3 characters, CCC format. The service will automatically pad the string up to 3 characters with spaces.	FFD 1.05 - No FFD 1.2 - No
identityDocumentCode	Identity document type code, 1245	2-character string, CC format.	FFD 1.05 - No FFD 1.2 - No
identityDocumentData	Data of the identity document, 1246	String up to 64 characters.	FFD 1.05 - No FFD 1.2 - No
address	Buyer (client) address, 1254	String from 1 to 239 characters.	FFD 1.05 - No FFD 1.2 - No

2.1.1. 9 Transaction details of the check , 1270

date	Date, time of operation, 1273	Time as a string in ISO 8601 format.	FFD 1.05 - No FFD 1.2 - No
id	Operation ID, 1271 *Takes the value "0" until the value of the attribute of the Federal Tax Service of Russia is determined.*	A number from 0 to 255.	FFD 1.05 - No FFD 1.2 - No
value	Transaction data, 1272	String from 1 to 64 characters.	FFD 1.05 - No FFD 1.2 - No

2.1.1. 10 Branch details of the check, 1261

foivId	FOIV ID, 1262	String from 1 to 3 characters.	FFD 1.05 - No FFD 1.2 - No
causeDocumentDate	Date of founding document, 1263	String of 10 characters, format DD.MM.YYYY.	FFD 1.05 - No FFD 1.2 - No
with auseDocumentNumber	Founding document number, 1264	String from 1 to 32 characters.	FFD 1.05 - No FFD 1.2 - No
value	Industry attribute value, 1265	String from 1 to 239 characters.	FFD 1.05 - No FFD 1.2 - No

2.1.1. eleven Barcodes of the subject of calculation

ean8	CT EAN-8, 1301	String 8 characters,	FFD 1.05 - No FFD 1.2 - No
ean13	CT EAN-13, 1302	String 13 characters,	FFD 1.05 - No FFD 1.2 - No
itf14	CT ITF-14, 1303	String 14 characters,	FFD 1.05 - No FFD 1.2 - No
gs1	CT GS1.0 , 1304	String from 1 to 38 characters,	FFD 1.05 - No FFD 1.2 - No
mi	CT MI, 1307	String 20 characters,	FFD 1.05 - No FFD 1.2 - No
egais20	CT EGAIS-2.0, 1308	String 23 characters,	FFD 1.05 - No FFD 1.2 - No
egais30	CT EGAIS-3.0, 1309	String 14 characters,	FFD 1.05 - No FFD 1.2 - No
f1	KT F.1 , 1320	String from 1 to 32 characters,	FFD 1.05 - No FFD 1.2 - No
f2	KT F. 2, 1321	String from 1 to 32 characters,	FFD 1.05 - No FFD 1.2 - No
f3	KT F. 3, 1322	String from 1 to 32 characters,	FFD 1.05 - No FFD 1.2 - No
f4	KT F. 4, 1323	String from 1 to 32 characters,	FFD 1.05 - No FFD 1.2 - No
f5	KT F. 5, 1324	String from 1 to 32 characters,	FFD 1.05 - No FFD 1.2 - No
f6	KT F. 6, 1325	String from 1 to 32 characters,	FFD 1.05 - No FFD 1.2 - No

2.1.1. 12 Fractional quantity of marked goods, 1291

Numerator	Numerator, 1293	Unsigned integer 8 bytes	FFD 1.05 - No FFD 1.2 - No
Denominator	Denominator, 1294	Unsigned integer 8 bytes	FFD 1.05 - No FFD 1.2 - No

In response: api can return the following http status codes

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

Example request :

```
{
  " id " : "12345678990" ,
  " inn " : "123456789012" ,
  " group " : "Main" ,
  " content " : {
    " type " : 1 ,
    " positions " : [
{
        " quantity " : 1.000 ,
        " price " : 123.45 ,
        " tax " : 6 ,
        " text " : " Bulka " ,
        " paymentMethodType " : 4 ,
        paymentSubjectType : 1 _ _
},
{
        " quantity " : 2.000 ,
        " price " : 4.45 ,
        " tax " : 4 ,
        " text " : " Matches " ,
        " paymentMethodType " : 3 ,
        paymentSubjectType : 13 _ _
}
],
    " checkClose " : {
      " payments " : [
{
        " type " : 1 ,
        " amount " : 123.45
},
{
        " type " : 2 ,
```

```json
      " amount " : 8.90000
   }
  ],
      " taxationSystem " : 1
 },
   " customerContact " : "foo@example.com"
  }
}
```

An example of a request with agent data, additional user details, supplier data, vending machine number, billing address and billing place:

```json
{
 " id " : "12345678990" ,
 " inn " : "123456789012" ,
 " group " : "Main" ,
 " key " : "1234567" ,
 " content " : {
  " type " : 1 ,
  " positions " : [
{
      " quantity " : 1.000 ,
      " price " : 123.45 ,
      " tax " : 6 ,
      " text " : " Bulka " ,
      " paymentMethodType " : 4 ,
      " paymentSubjectType " : 1 ,
      " nomenclatureCode " : "igQVAAADMTIzNDU2Nzg5MDEyMwAAAAAAQ==" ,
      " agentType " : 127 ,
      " agentInfo " : {
        " paymentTransferOperatorPhoneNumbers " : [ "+79200000001" , "+74997870001" ],
```

```json
        " paymentAgentOperation " : " Some operation 1" ,
        " paymentAgentPhoneNumbers " : [ "+79200000003" ],
        " paymentOperatorPhoneNumbers " : [ "+79200000002" , "+74997870002" ],
        " paymentOperatorName " : " OOO \" Atlant \"" ,
        " paymentOperatorAddress " : " Voronezh , Nedogonnaya st . , 84 " ,
        " paymentOperatorINN " : "7727257386"
},
      " unitOfMeasurement " : " kg " ,
      " additionalAttribute " : "Add. attribute And all here" ,
      " manufacturerCountryCode " : "643" ,
      " customsDeclarationNumber " : " AD 11/77 of 08/01/2018 " ,
      excise : 23.45 _ _
},
{
      " quantity " : 2.000 ,
      " price " : 4.45 ,
      " tax " : 4 ,
      " text " : " Matches " ,
      " paymentMethodType " : 3 ,
      " paymentSubjectType " : 13 ,
      " supplierINN " : "9715225506" ,
      " supplierInfo " : {
        " phoneNumbers " : [ "+79266660011" , "+79266660022" ],
        "name" : " PAO \" Adamas \""
}
}
```

```json
    ],
    " checkClose " : {
      " payments " : [
{
        " type " : 1 ,
        " amount " : 123.45
},
{
        " type " : 2 ,
        " amount " : 8.90000
}
],
      " taxationSystem " : 1
},
    " customerContact " : "foo@example.com" ,
    " agentType " : 127 ,
    " paymentTransferOperatorPhoneNumbers " : [ "+79260000001" , "+74957870001" ],
    " paymentAgentOperation " : " Some operation " ,
    " paymentAgentPhoneNumbers " : [ "+79260000003" ],
    " paymentOperatorPhoneNumbers " : [ "+79260000002" , "+74957870002" ],
    " paymentOperatorName " : " Ltd. \" Dewdrop \"" ,
    " paymentOperatorAddress " : " Moscow , Masterkova 4" ,
    " paymentOperatorINN " : "9715225506" ,
    " supplierPhoneNumbers " : [ "+74957870004" ],
    " additionalUserAttribute " : {
      " name " : "Darling quote" ,
```

```
        "value" : "A healthy mind in a healthy body, this slogan has not died out yet!"

    },

    " automatNumber " : "123456789" ,

    " settlementAddress " : " Moscow , Krasnaya area , d .1" ,

    " settlementPlace " : " Ward #6" ,

    " additionalAttribute " : " Additional attribute check ,

    " customer " : "Kuznetsov Ivan Petrovich" ,

    " customerINN " : "789456123488"

}

}
```

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

```
{
  " id " : "12345678990" ,
  " inn " : "7725327863" ,
  group : "Main_2" ,
  " key " : "1234567" ,
  " content " : {
    " type " : 1 ,
    " position " : [
      {
        " quantity " : 1.000
        " price " : 123.45 ,
        " tax " : 6 ,
        " text " : " Bulka " _
        paymentMethodType : 4 ,
        paymentSubjectType : 1 ,
        agentType : 127
        " agentInfo " : {
          " paymentTransferOperatorPhoneNumbers " : [ "+79200000001" , "+74997870001" ],
          paymentAgentOperation : " Some kind operation 1" ,
          " paymentAgentPhoneNumbers " : [ "+79200000003" ],
          " paymentOperatorPhoneNumbers " : [ "+79200000002" , "+74997870002" ],
          paymentOperatorName : " LLC \" Atlanta \" " ,
          " paymentOperatorAddress " : " Voronezh , st . Nedogonnaya , d . 84" ,
```

```
        " paymentOperatorINN " : "7727257386"
      },
      " additionalAttribute " : " Additional attribute  And all here " ,
      " manufacturerCountryCode " : "643" ,
      " customsDeclarationNumber " : " AD 11/77 of 08/01/2018 " ,
      " excise " : 23.45 ,
      " unitTaxSum " : 0.23 ,
      " itemCode "
: "010460406000600021N4N57RSCBUZTQ\u001d2403004002910161218\u001d1724010191ffd0\u001d92tIAF/YVoU4roQS3M/m4z78yFq0fc/WsSmLeX5QkF/YVWwy8IMYAeiQ91X
a2z/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 " : " Matches "
      paymentMethodType : 3 ,
```

```json
      paymentSubjectType : 13 ,
      " supplierINN " : "9715225506" ,
      supplierInfo : {
        " phoneNumbers " : [ "+79266660011" , "+79266660022" ],
        "name" : " PAO \" Adamas \" "
      },
      " quantityMeasurementUnit " : 10
    }
  ],
  " checkClose " : {
    " payments " : [
      {
        " type " : 1 ,
        " amount " : 123.45
      },
      {
        " type " : 2 ,
        " amount " : 8.90000
      }
    ],
    " taxationSystem " : one
  },
  " customerContact " : "foo@example.com" ,
  " additionalUserAttribute " : {
    " name " : " Favorite Quote"
    "value" : "A healthy mind in a healthy body, this slogan has not gone out yet!"
  },
  automatNumber : "123456789" ,
  settlementAddress : " Moscow , Krasnaya area , d .1" ,
  settlementPlace : " Ward #6" ,
  " additionalAttribute " : " Extra attribute check ,
  cashier : " Cashier "
  senderEmail : " en@example.mail " ,
  " customer info " : {
    " name " : " Kuznetsov Ivan Petrovich ,
    " inn " : "7725327863" ,
    birthDate : "15.09.1988" ,
    " citizenship " : "643" ,
    " identityDocumentCode " : "01" ,
    " identityDocumentData " : " multipassport " ,
    " address " : " Baseennaya 36"
  },
  " operationAttribute " : {
```

```
      " 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 with request processing errors

errors	Array of request processing errors	Array of strings

An example of an error response:

```
{
  " errors " : [
    "Document ID ' Id ' not specified " ,
    "TIN of organization 'INN' is not specified" ,
    "Missing content of document ' Content '"
  ]
}
```

2.2 Check status

Request : **GET /api/v2/documents/ {inn} /status /{ document_id }**

{ inn } – TIN of the organization for which the check is made

{ document _ id } – ID of the document that was specified when it was created

This request does not use a SHA 256 - RSA signature .

Answer: api can return the following status codes

- 202 Accepted - the check has been created and added to the queue for processing, but has not yet been processed, an empty response body
- 400 Bad Request - the organization was not found, the check with the specified identifier was not found
- 401 Unauthorized - The client certificate has not been validated
- 200 OK – receipt processed, response body p.2.2.1
- 524 Document Expired Before Processing - the server failed to process the document in the allotted time, send a check with a new identifier for re-processing, an empty response body
- 422, Unprocessable Entity - the document cannot be processed due to CM check errors, response body p.2.2.2

2.2.1 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
fs Number	Fiscal accumulator number	String 16 characters
ofdName	Name of OFD	String up to 256 characters
odfwebsite	CRF website	Line up to 58? characters

odfINN	TIN OFD	String 12 characters
fnsWebsite	Website of the Federal Tax Service	String up to 256 characters
companyINN	User TIN	String 12 characters
withcompanyName	User name	String up to 256 characters
DocumentNumber	FD number	Number
sshiftNumber	shift number	Number
documentIndex	Check number for shift	Number
processedAt	Time of registration of a fiscal document in the Federal Tax Service	Time as a string in ISO 8601 format
content	Document content	Structure of clause 2.1.1.1
change	Change	Decimal number up to 2 characters after the dot
fp	fiscal sign	String 10 characters
callbackUrl	URL for sending check processing results by POST request	String from 1 to 1024 characters or null
meta	Request metadata	String from 1 to 128 characters or null

Example response :

```
{
  " id " : "12345678990" ,
  " deviceSN " : "0000000000001358" ,
  " deviceRN " : "0000000400054952" ,
  " fsNumber " : "9999078900001341" ,
  " ofdName " : " LLC \" Tier \" ( \" OFD - I \" )" ,
  " ofdWebsite " : "www.ofd-ya.ru" ,
  " ofdinn " : "7728699517" ,
  " fnsWebsite " : "www.nalog.ru" ,
  " companyINN " : "123456789012" ,
  " companyName " : " CJSC TRADE OBJECT №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 " : " Bulka " ,
            " paymentMethodType " : 4 ,
            paymentSubjectType : 1
    },
    {
            " quantity " : 2.000 ,
            " price " : 4.45 ,
            " tax " : 4 ,
            " text " : " Matches " ,
            " 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	Structure array 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 CM check result

fsCheckStatus	Result of KM check, 2004	Number from 0 to 255
fsCheckStatusCause	The reason why the local check was not carried out: 0 - CM checked in FN 1 - CM of this type is not subject to verification in the FN 2 - FN does not contain a verification key for the verification code of this CM 3 - Verification is not possible because GS1 Application IDs 91 and/or 92 are missing or their format is incorrect. 4 - Checking the CM in the FN is impossible for another reason	Number from 0 to 255
fsItemCodeType	Marking code type, 2100	Number from 0 to 255
checkResult	Product details check result, 2106	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 sign of the answer; 2 - Wrong format of response details ; 3 - Invalid request number in the response; 4 - Wrong FN number; 5 - Invalid CRC of the data block; 7 - Invalid response length. The value is 0xFF if the server did not respond within the timeout.	Number from 0 to 255
oismResponse	CMIS response to the request. The field is absent ifoismCheckResultCode is different from 0 or the checkout is offline.	Structure p.2. 2.2.3,

2.2.2.3 Server response to CM check request

dateTime	Request date and time, 2114	Time as a string in ISO 8601 format
resultCode	Request handling codes, 2105	Number from 0 to 255
itemCodeType	Marking code type, 2100	Number from 0 to 255,
oismStatus	CMIS response on item status, 2109	Number from 0 to 255,
processingResults	Request processing results, 2005	Number from 0 to 255,
itemId	Item ID, 2101	String from 1 to 38 characters,

Example 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 is an asynchronous request; after its execution, the receipt is queued for processing. In addition to the check data, the client sends a unique document identifier, using which later, he will be able to request the status of the correction check (clause 2.4). Plus, this identifier ensures the idempotency of queries.

The identifier must be unique within the organization.

The TIN is used to determine the device with which a check can be processed.

Group in the request may be null or not specified, 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	Document ID	String from 1 to 64 characters.	The parameter is required.
Inn	TIN of the organization for which the check is made	A string of 10 or 12 characters .	The parameter is required.
group	Group of devices with which the check will be punched	String from 1 to 32 characters or null	The parameter is required.
content	Document content	Structure of clause 2.3.1.1	
key	The name of the key to be used to verify the signature. For clients, their TIN is used, for partners and paying agents, the code with a mask is 301 ****, for vending 401 ****.	String from 1 to 32 characters, or null .	The parameter is required.
callbackUrl	URL for sending check processing results by POST request	String from 1 to 1024 characters, or null .	The parameter is optional.
callbackApiKey	API key for calling the callback	String from 1 to 3072 characters or null	The parameter is optional
meta	Request metadata	String from 1 to 128 characters, or null .	The parameter is optional.

2.3.1.1 Document content

correctionType	Correction type 1173:
	    0. Self
	    1. By prescription
		Number.	The parameter is required.

type	Settlement sign, 1054:
	    1. Arrival
	    3. Consumption
		Number .	The parameter is required.

causeDocumentDate	The document date of the basis for the correction is 1178. In this attribute, always indicate the time as 00:00:00	Time as a string in ISO 8601 format.	The parameter is required.
with auseDocumentNumber	The number of the order of the tax authority, on the basis of which this correction is carried out, 1179	String from 1 to 32 characters.	The parameter is required.
totalSum	The amount of the calculation indicated in the check (BSO), 1020	Decimal number accurate to 2 characters after the dot .	The parameter is required.
CashSum	Check amount (BSO) in cash, 1031	Decimal number up to 2 characters after the dot	
eCashSum	Check amount (BSO) non-cash, 1081	Decimal number up to 2 characters after the dot	
prepaymentSum	Check amount (BSO) prepaid (offset of advance payment and (or) previous payments), 1215	Decimal number up to 2 characters after the dot	
postpaymentSum	Check amount (SSO) postpaid (on credit), 1216	Decimal number up to 2 characters after the dot	
othePaymentTypeSum	Amount on check (BSO) counter provision, 1217	Decimal number up to 2 characters after the dot	
tax1Sum	The amount of the VAT check at the rate of 20%, 1102	Decimal number up to 2 characters after the dot	
tax2Sum	The amount of VAT check at a rate of 10%, 1103	Decimal number up to 2 characters after the dot	
tax3Sum	Settlement amount on a check with VAT at a rate of 0%, 1104	Decimal number up to 2 characters after the dot	
tax4Sum	Amount of check settlement without VAT, 1105	Decimal number up to 2 characters after the dot	
tax5Sum	The amount of the VAT check according to calc . rate 20/120, 1106	Decimal number up to 2 characters after the dot	
tax6Sum	The amount of the VAT check according to calc . rate 10/110, 1107	Decimal number up to 2 characters after the dot	

tax7Sum	The amount of the VAT check at the rate of 5%	Decimal number up to 2 characters after the dot	
tax8Sum	The amount of the VAT check at the rate of 7%	Decimal number up to 2 characters after the dot	
tax9Sum	The amount of the VAT check according to calc . rate 5/105	Decimal number up to 2 characters after the dot	
tax10Sum	The amount of the VAT check according to calc . rate 7/107	Decimal number up to 2 characters after the dot	

taxationSystem	Applicable system of taxation, 1055:
	    0. General
	    1. Simplified income
	    2. Simplified income minus expenses
	    3. Single tax on imputed income
	    4. Single agricultural tax
	5. Patent taxation system
		Number.	The parameter is required.

automatNumber	Machine number, 1036	A string with a length of 1 to 20 characters, a mandatory field if the group has the sign of transferring machine data, otherwise it should not be transferred. **Only for vending and transport.**	The parameter is required only for vending and transport.
settlementAddress	Settlement address, 1009	A string from 1 to 243 characters long, a mandatory field if the group has the automaton data transmission flag, otherwise it should not be transmitted. **Only for vending and transport.**	The parameter is required only for vending and transport.
settlementPlace	Place of settlement, 1187	A string from 1 to 243 characters long, a mandatory field if the group has the automaton data transmission flag, otherwise it should not be	The parameter is required only for vending and transport.

		transmitted. **Only for vending and transport.**	

In response: api can return the following http status codes

- 201 Created – check has been created and added to the processing queue, empty response body
- 401 Unauthorized - The client certificate has not been validated
- 409 Conflict - a check with the given 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 the verification, the response body p.2.1.2
- 503 Service Unavailable - the document queue is full, the header is returned in the response Retry - After with a timeout in seconds, after which it is worth repeating the request, response body p.2.1.2.

2.3.2 Response body with request processing errors

errors	Array of request processing errors	Array of strings

Example request :

```json
{
  " 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 " : " Moscow , Krasnaya _ _ area , d .1" ,

    settlementPlace : _ _ "Ward No. 6" ,


  }
}
```

An example of an error response:

```
{
  " errors " : [
    "Document ID ' Id ' not specified " ,
    "TIN of organization 'INN' is not specified" ,
    "Missing content of document ' Content '"
]
}
```

2.4 Correction check status

Request : **GET /api/v2/corrections/{inn}/status/{ document_id }**

{ inn } – TIN of the organization for which the check is made

{ document _ id } – ID of the document that was specified when it was created

This request does not use a SHA 256 - RSA signature .

Answer: api can return the following status codes

- 202 Accepted - the check has been created and added to the queue for processing, but has not yet been processed, an empty response body
- 400 Bad Request – the organization was not found, the check with the specified identifier was not found
- 401 Unauthorized - The client certificate has not been validated
- 200 OK – receipt processed, response body p.2.4.1

2.4.1 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	Name of OFD	String up to 256 characters
odfwebsite	CRF website	String up to 58 characters
odfINN	TIN OFD	String 12 characters
fnsWebsite	Website of the Federal Tax Service	String up to 256 characters
companyINN	User TIN	String 12 characters

companyName	User name	String up to 256 characters
DocumentNumber	FD number	Number
shiftNumber	shift number	Number
documentIndex	Check number for shift	Number
processedAt	Time of registration of a fiscal document in the Federal Tax Service	Time as a string in ISO 8601 format
content	Document content	Structure of clause 2.3.1.1
fp	fiscal sign	String 10 characters
callbackUrl	URL for sending check processing results by POST request	String from 1 to 1024 characters or null
meta	Request metadata	String from 1 to 128 characters or null

Example response :

```
{
  " id " : "12345678990" ,
  " deviceSN " : "0000000000001358" ,
  " deviceRN " : "0000000400054952" ,
  " fsNumber " : "9999078900001341" ,
  " ofdName " : " LLC \" Tier \" ( \" OFD - I \" )" ,
  " ofdWebsite " : "www.ofd-ya.ru" ,
  " ofdinn " : "7728699517" ,
  " fnsWebsite " : "www.nalog.ru" ,
  " companyINN " : "123456789012" ,
  " companyName " : " CJSC TRADE OBJECT №1" ,
  " documentNumber " : 117 ,
  " shiftNumber " : 20 ,
  " documentIndex " : 5 ,
  " processedAt " : "2017-02-14T14:16:00" ,
  " content " : {
    " correctionType " : 1 ,
    " type " : 1 ,
    " description " : " NOT I WANT TO BUT NEED " ,
    " causeDocumentDate " : "2017-08-10T00:00:00" ,
    " causeDocumentNumber " : " ФЗ -54" ,
    " totalSum " : 17.25 ,
    " cashSum " : 1.23 ,
    " eCashSum " : 2.34 ,
    " prepaymentSum " : 5.67 ,
    " postpaymentSum " : 4.56 ,
```

```json
      " 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 Status KKT in group

Request : **GET /api/v2/ devices/status/{inn}/{ group_name }**

{ inn } – TIN of the organization

{ group _ name } – device group name

This request does not use a SHA 256 - RSA signature .

Answer: api can return the following status codes

- 400 Bad Request - organization or group not found, response body 2.5.2
- 401 Unauthorized - The client certificate has not been validated
- 200 OK - no errors, response body p.2.5.1

2.5.1 Response Body

devices	List of devices in a group	Array of structures p.2.5.1.1

2.5.1.1 Device information object structure

deviceSN	Device serial number	String up to 20 characters	Obligatory field
fsNumber	Fiscal accumulator number	String 16 characters,	Obligatory field
ofdName	Name of OFD	String up to 256 characters,	optional field
odfAddress	Address / IP and port of sending checks to OFD	String up to 64 characters,	optional field
unsentDocumentsCount	Number of unsent documents	Number	Obligatory field
firstUnsentDocumentDate	Date and time when the first unsent document was created	Time as a string in ISO 8601 format,	optional field

fsDocumentsCount	The total number of documents in the FN	Number	Obligatory field
fsExpirationDate	FN expiration date	Time as a string in ISO 8601 format,	optional field

2.5.2 Response body with request processing errors

errors	Array of request processing errors	Array of strings

Answer example:

```json
{
  " devices " : [
{
    " deviceSN " : "0000000000001358" ,
    " fsNumber " : "9999078900001341" ,
    " ofdName " : "OOO \" Yarus \" ( \" OFD-I \" )" ,
    " 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 Verification parameters access

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

Request : **POST /api/v2/check/**

2.6.1 Request body

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

In response: api can return the following http status codes

- 200 OK - request completed successfully
- 401 Unauthorized - The client certificate has not been validated
- 400 Bad Request - the data has an incorrect format or the signature did not pass the verification, the response body p.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 " : [
    "Signature failed verification"
]
}
```

67

2. 7 Creation of FFD correction check 1.2

Request : **POST /api/v2/ correction12 /**

2.7.1 Request body

id	Document ID	String from 1 to 64 characters.	The parameter is required.
inn	TIN of the organization for which the check is made	A string of 10 or 12 characters.	The parameter is required.
group	Group of devices with which the check will be punched	String from 1 to 32 characters, or null .	The parameter is required.
content	Document content	Structure p.2. 7.1.1	
key	The name of the key to be used to verify the signature. For clients, their TIN is used, for partners and paying agents, the code with a mask is 301 ****, for vending 401 ****.	String from 1 to 32 characters.	The parameter is required.
callbackUrl	URL for sending check processing results by POST request	String from 1 to 1024 characters, or null .	optional field.
callbackApiKey	API key for calling the callback	String from 1 to 3072 characters or null	The parameter is optional
meta	Request metadata	String from 1 to 128 characters, or null .	optional field.
ignoreItemCodeCheck	Flag indicating whether the CM check should be ignored. If the flag is not specified, then to generate a check, all CMs must successfully pass the check: 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 status 422 Unprocessable entity	Boolean value The default value is false .	optional field.

2. 7.1.1 Document content

Attribute	Description	A comment	obligatory
ffdVersion	FFD version number, 1209 4 - FFD 1.2	Number.	required parameter
type	Settlement sign, 1054: 1. Coming 2. return parish 3. Consumption 4. Expense return	Number.	required parameter
positions	List of calculation items, 1059	Array of structures p.2.1.1.2	
checkClose	Receipt closing options	Structure of clause 2.1.1.3	
customerContact	Phone or email address of the buyer, 1008	String from 1 to 64 characters, format +{C} or {C}@{C}, {none}.	required parameter
correctionType	Correction type 1173: 0. Self 1. By prescription	Number.	required parameter
causeDocumentDate	The document date of the basis for the correction is 1178. In this attribute, always indicate the time as 00:00:00	Time as a string in ISO 8601 format.	required parameter
CauseDocumentNumber	The number of the order of the tax authority, on the basis of which this correction is carried out, 1179	String from 1 to 32 characters.	required parameter (If a check is generated with the correction type "By prescription").
additionalUserAttribute	Additional username, 1084	The structure of clause 2.1.1. five	optional field
additionalAttribute	Additional details of the check (BSO) , 119 2	String from 1 to 16 characters	optional field
automatNumber	Machine number, 1036	A string with a length of 1 to 20 characters, a mandatory field if the group has the sign of transferring machine data, otherwise it should not be transferred. **Only for vending and transport.**	This parameter is required only for vending and transport.
settlementAddress	Settlement address, 1009	A string from 1 to 243 characters long, a mandatory field if the	The parameter is required only for vending and transport.

		group has the automaton data transmission flag, otherwise it should not be transmitted. **Only for vending and transport.**	
settlementPlace	Place of settlement, 1187	A string from 1 to 243 characters long, a mandatory field if the group has the automaton data transmission flag, otherwise it should not be transmitted. **Only for vending and transport.**	The parameter is required only for vending and transport.
cashier	Cashier, 1021	A string from 1 to 64 characters, a mandatory field if the group has the sign of passing courier parameters, otherwise it should not be sent. **For couriers only.**	This parameter is required only for couriers.
cashierINN	TIN of the cashier, 1203	A string of 12 characters, format DDDDDDDDDDDD,	optional field
senderEmail	Check sender's email address, 1117	String from 1 to 64 characters, format +{C} or {C}@{C}, {none}.	optional field
totalSum	The amount of the calculation indicated in the check (BSO), 1020.

Fiscal Document Formats allow rounding down in this field. Literally, the FFD requirement sounds like this: "The value in rubles, excluding kopecks, must be equal to the value of the sum of all details "the cost of the subject of calculation, taking into account discounts and markups" (tag 1043) in rubles, excluding kopecks." | Decimal number with an accuracy of 2 characters after the point, | optional field |

	To avoid inconsistencies, we recommend using the exact value without rounding.		
vat1Sum	The amount of the VAT check at the rate of 20%, 1102	Decimal number with an accuracy of 2 characters after the point,	optional field
vat2Sum	The amount of VAT check at a rate of 10%, 1103	Decimal number with an accuracy of 2 characters after the point,	optional field
vat3Sum	Settlement amount on a check with VAT at a rate of 0%, 1104	Decimal number with an accuracy of 2 characters after the point,	optional field
vat4Sum	Amount of check settlement without VAT, 1105	Decimal number with an accuracy of 2 characters after the point,	optional field
vat5Sum	The amount of the VAT check according to calc . rate 20/120, 1106	Decimal number with an accuracy of 2 characters after the point,	optional field
vat6Sum	The amount of the VAT check according to calc . rate 10/110, 1107	Decimal number with an accuracy of 2 characters after the point,	optional field
vat7Sum	The amount of the VAT check at the rate of **5%**	Decimal number with an accuracy of 2 characters after the point,	optional field
vat8Sum	The amount of the VAT check at the rate of 7%	Decimal number with an accuracy of 2 characters after the point,	optional field
vat9Sum	The amount of the VAT check according to calc . rate 5/105	Decimal number with an accuracy of 2 characters after the point,	optional field
vat10Sum	The amount of the VAT check according to calc . rate 7/107	Decimal number with an accuracy of 2 characters after the point,	optional field
customer info	Information about the buyer (client), 1256	The structure of clause 2.1.1.8,	optional field
operationalAttribute	Operating details of the check, 1270	The structure of clause 2.1.1.9,	optional field
industryAttribute	Branch requisite of the check, 1261	Structure p.2.1.1.10,	optional field

Note: for historical reasons, the numbering of tax rates **vat1Sum, vat2Sum, vat3Sum, vat4Sum, vat5Sum, vat6Sum, vat7Sum, vat8Sum, vat9Sum, vat10Sum** in OrangeData formats is slightly different from the numbering of tax rates in the FFD for tag 1199. When developing integration, one should strictly adhere to this manual.

In response: api can return the following http status codes

- 201 Created – check has been created and added to the processing queue, empty response body
- 401 Unauthorized - the client certificate has not been validated
- 409 Conflict - a check with the given 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 the verification, the response body p.2.1.2
- 503 Service Unavailable - the document queue is full, the header is returned in the response Retry - After with a timeout in seconds, after which it is worth repeating the request, response body p.2.7.2.

Example request :

```json
{
  " id " : "12345678990" ,
  " inn " : "7725327863" ,
  group : "Main_2" ,
  " key " : "1234567" ,
  " content " : {
    " type " : 1 ,
    " position " : [
      {
        " quantity " : 1.000
        " price " : 123.45 ,
        " tax " : 6 ,
        " text " : " Bulka "
        paymentMethodType : 4 ,
        paymentSubjectType : 1 ,
        agentType : 127
        " agentInfo " : {
          " paymentTransferOperatorPhoneNumbers " : [ "+79200000001" , "+74997870001" ],
          paymentAgentOperation : " Some kind operation 1" ,
          " paymentAgentPhoneNumbers " : [ "+79200000003" ],
          " paymentOperatorPhoneNumbers " : [ "+79200000002" , "+74997870002" ],
          paymentOperatorName : " LLC \" Atlanta \" " ,
          " paymentOperatorAddress " : " Voronezh , st . Nedogonnaya , d . 84" ,
          " paymentOperatorINN " : "7727257386"
        },
        " additionalAttribute " : " Additional attribute _ And all here " ,
        " manufacturerCountryCode " : "643" ,
```

```json
      " customsDeclarationNumber " : " AD 11/77 of 08/01/2018 " ,
      " excise " : 23.45 ,
      " unitTaxSum " : 0.23 ,
      " itemCode " : "0104604060006000021N4N57RSCBUZTQ\u001d2403004002910161218\u001d
1724010191ffd0\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 " : " Matches "
      paymentMethodType : 3 ,
      paymentSubjectType : 13 ,
      " supplierINN " : "9715225506" ,
      supplierInfo : {
      " phoneNumbers " : [ "+79266660011" , "+79266660022" ],
      "name" : " PAO \" Adamas \" "
      },
      " quantityMeasurementUnit " : 10
    }
  ],
  " checkClose " : {
  " payments " : [
```

```json
      {
        " type " : 1 ,
        " amount " : 123.45
      },
      {
        " type " : 2 ,
        " amount " : 8.90000
      }
    ],
    " taxationSystem " : one
  },
  " customerContact " : "foo@example.com" ,
  " correctionType " : 1 ,
  " causeDocumentDate " : "2021-08-13T00:00:00" ,
  causeDocumentNumber : "42/666" ,
  " additionalUserAttribute " : {
    " name " : "Darling quote" ,
    "value" : "A healthy mind in a healthy body, this slogan has not gone out yet!"
  },
  automatNumber : "123456789" ,
  settlementAddress : " Moscow , Krasnaya area , d .1" ,
  settlementPlace : " Ward #6" ,
  " additionalAttribute " : " Extra attribute check ,
  cashier : " Cashier "
  senderEmail : " en@example.mail " ,
  " customer info " : {
    " name " : " Kuznetsov Ivan Petrovich ,
    birthDate : "15.09.1988" ,
    " citizenship " : "643" ,
    " identityDocumentCode " : "01" ,
    " identityDocumentData " : " multipassport " ,
    " address " : " Baseennaya 36"
  },
  " operationAttribute " : {
    " date " : "2021-08-12T18:36:16" ,
    " id " : 0 ,
    " value " : "operational"
  },
  " industryAttribute " : {
    " foivId " : "010" ,
    " causeDocumentDate " : "11.08.2021" ,
    causeDocumentNumber : "999" ,
    " value " : "industry"
```

74

```
    },
    " 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 " : [
    "Document ID ' Id ' not specified " ,
    "TIN of organization 'INN' is not specified" ,
    "Missing content of document ' Content '"
  ]
}
```

2.8 FFD correction check status 1.2

Request : **GET /api/v2/correction12/{inn}/status/{ document_id }**

{ inn } – TIN of the organization for which the check is made

{ document _ id } – ID of the document that was specified when it was created

This request does not use a SHA 256 - RSA signature .

Answer: api can return the following status codes

- 202 Accepted - the check has been created and added to the queue for processing, but has not yet been processed, an empty response body
- 400 Bad Request - the organization was not found, the check with the specified identifier was not found
- 401 Unauthorized - The client certificate has not been validated
- 200 OK – receipt processed, response body p.2.11.1
- 524 Document Expired Before Processing - the server failed to process the document in the allotted time, send a check with a new identifier for re-processing, an empty response body
- 422, Unprocessable Entity - the document cannot be processed due to CM check errors, response body p.2.2.2

2.8.1 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	Name of OFD	String up to 256 characters
odfWebsite	CRF website	Line up to 58 symbols

odfINN	TIN OFD	String 12 characters
fnsWebsite	Website of the Federal Tax Service	String up to 256 characters
companyINN	User TIN	String 12 characters
companyName	User name	String up to 256 characters
DocumentNumber	FD number	Number
shiftNumber	shift number	Number
documentIndex	Check number for shift	Number
processedAt	Time of registration of a fiscal document in the Federal Tax Service	Time as a string in ISO 8601 format
content	Document content	Structure p.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 check processing results by POST request	String from 1 to 1024 characters or null
meta	Request metadata	String from 1 to 128 characters or null

Example response :

```
{
  " id " : "12345678990" ,
  " deviceSN " : "0000000000001358" ,
  " deviceRN " : "0000000400054952" ,
  " fsNumber " : "9999078900001341" ,
  " ofdName " : " LLC \" Tier \" ( \" OFD - I \" )" ,
  " ofdWebsite " : "www.ofd-ya.ru" ,
  " ofdinn " : "7728699517" ,
  " fnsWebsite " : "www.nalog.ru" ,
  " companyINN " : "123456789012" ,
  " companyName " : " CJSC TRADE OBJECT №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 " : " Bulka " ,
        " paymentMethodType " : 4 ,
        paymentSubjectType : 1
},
{
        " quantity " : 2.000 ,
        " price " : 4.45 ,
        " tax " : 4 ,
        " text " : " Matches " ,
        " 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.	The parameter is required.
inn	TIN of the organization for which the request is sent	A string of 10 or 12 characters.	The parameter is required.
group	Group of devices that will process the request	String from 1 to 32 characters or null	The parameter is required.
content	Document content	Structure p.2. 9.1.1	
key	The name of the key to be used to verify the signature. For clients, their TIN is used, for partners and paying agents, the code with a mask is 301 ****, for vending 401 ****.	String from 1 to 32 characters.	The parameter is required.
callbackUrl	URL for sending the results of POST validation processing by request	String from 1 to 1024 characters, or null .	Optional field.
callbackApiKey	API key for calling the callback	String from 1 to 3072 characters or null	The parameter is optional
meta	Request metadata	String from 1 to 128 characters, or null .	Optional field.

2.9.1.1 Document content

Attribute	Description	A comment	obligatory
planned status	Planned status, 2003: 1 Piece goods subject to mandatory labeling with an identification tool have been sold,	A number from 0 to 256.	The parameter is required.

	2 Measured goods subject to mandatory labeling with an identification tool, in the implementation stage, 3 Piece goods subject to mandatory marking with an identification tool are returned, 4 Part of the goods subject to mandatory marking with an identification tool is returned, 255 The status of goods subject to mandatory marking with an identification tool has not changed		
ItemCode	Marking code, 2000 The KM itself, as it was read by the scanner	String from 1 to 223 characters.	The parameter is required.
quantityMeasurementUnit	Measure of the quantity of the subject of calculation, 2108	A number from 0 to 255.	optional field
quantity	Number of subject of calculation, 1023	Decimal number up to 6 characters after the dot.	optional field
fractionalQuantity	Fractional quantity of marked goods, 1291	The structure of clause 2.1.1.12.	optional field

In response: api can return the following http status codes

- 201 Created – check has been created and added to the processing queue, empty response body
- 401 Unauthorized - The client certificate has not been validated
- 409 Conflict - a check with the given 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 the verification, the response body p.2.9.2
- 503 Service Unavailable - the document queue is full, the header is returned in the response Retry - After with a timeout in seconds, after which it is worth repeating the request, response body p.2.9.2.

Example request :

```json
{
  " plannedStatus " : 1 ,
  " itemCode " : "00000046210654eK7fYtcAAModGVz"
}
```

```json
{
  " plannedStatus " : 2 ,
  " itemCode " : "00000046210654eK7fYtcAAModGVz" ,
  " quantityMeasurementUnit " : 21 ,
  " quantity " : 1.234
}
```

```json
{
  " plannedStatus " : 4 ,
  " itemCode " : "00000046210654eK7fYtcAAModGVz" ,
  " quantityMeasurementUnit " : 0 ,
  " quantity " : one,
  " fractionalQuantity " : {
    " numerator " : 1 ,
    denominator : _ _ 2
  }
}
```

2.9.2 Body response from mistakes processing request

errors	Array of request processing errors	Array of strings

An example of an error response:

```json
{
  " errors " : [
    "Document ID ' Id ' not specified " ,
    "TIN of organization 'INN' is not specified" ,
    "Missing content of document ' Content '"
  ]
}
```

}

2. 10 Marking code verification status

Request : **GET /api/v2/ itemcode /{inn}/status/{ document_id }**

{ inn } – TIN of the organization for which the check is made

{ document _ id } – ID of the document that was specified when it was created

This request does not use a SHA 256 - 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 yet been processed, an empty response body
- 400 Bad Request - organization not found, request with specified ID not found
- 401 Unauthorized - The client certificate has not been validated
- 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
fs Number	Fiscal accumulator number	String 16 characters
processedAt	Time of registration of a fiscal document in the KM server	Time as a string in ISO 8601 format
content	Document content	Structure of clause 2.9.1.1
checkResult	Result of checking	Structure p.2. 2 . 2.2

callbackUrl	URL for sending check processing results by POST request	String from 1 to 1024 characters or null
meta	Request metadata	String from 1 to 128 characters or null

Answer example:

```
{
  " id " : "006 f 14 f 6 d 6 d 048 c 0859 c 0 f 0 bafa 1389 c " ,
  " 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
    }
  }
}
```

3. Swagger

api description available in form OpenAPI Specification(OAS).

By url **/swagger** swagger- ui available

By url **/swagger/v2/ swagger.json** - specification .

4. Links to the check.

This functionality is enabled in the Personal Account on the "Settings" page, the item "Allow clients to access receipts".

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

Generating mask: https:// **{ url }/{ inn }** / **{ document_id }**

{ url } – Service address for viewing receipts.

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

{ inn } – TIN of the organization for which the check is made

{ document_id } - ID of the document that was specified when it was created

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

Document Changes

Version 0.1.0 from 02/13/2017

First version .

Version 0.2.0 from 02/17/2017

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

Version 0.3.0 from 02/21/2017

Changed the time that is issued as a result of a receipt being broken into local, clarification on the price of the goods, a sign of sending to the OFD and receipt data has been added.

Version 0.4.0 from 03/23/2017

Values for the tax system and payment type parameters have been added, the group attribute has been added to the registration request.

Version 0.5.0 from 04/13/2017

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

Version 2.0.0 from 05/24/2017

Added TIN to the document status request.

API version changed to v2.

Version 2.1.0 from 05/29/2017

Added clarification about the default group.

Added mention of swagger .

Version 2.2.0 from 06/01/2017

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

Version 2.3.0 from 06/07/2017

Added description for 401 errors.

Added an example of signature generation for . net full .

Version 2.4.0 from 06/14/2017

Added an example of signature generation in php .

Version 2.5.0 from 06/19/2017

The message format has been changed to match what the API returns.

Added an example of a response with an error.

Version 2.6.0 from 06/20/2017

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

Version 2.7.0 from 06/26/2017

The example of a message with a signature has been changed.

Version 2.7.1 from 06/28/2017

Fixed the response code for a document not found when checking the status in accordance with the behavior of the system.

Version 2.8.0 from 07/31/2017

Added information about 503 error when sending a document.

Added parameters of the document position "Attribute of the subject of calculation" and "Attribute of the method of calculation".

The maximum amounts of the position and payment have been specified.

Increased quantity precision from 3 to 6 characters.

Where possible, the terminology used in the FFD has been replaced.

Version 2.9.0 from 08/03/2017

Added information about FFD version.

Points to which it is worth paying attention during development are described.

Version 2.9.1 from 08/08/2017

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

Version 2.10.0 from 08/10/2017

Added the ability to transfer agency data.

Added tag numbers corresponding to receipt data.

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

Version 2.12.0 from 08/30/2017

Added the ability to specify the key that was signed.

Version 2.13.0 from 10/11/2017

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

Version 2.14.0 from 11/09/2017

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

Version 2.14.1 dated 11/21/2017

Added clarification on the restriction on the maximum price per unit of the subject of calculation and the maximum quantity of the subject of calculation .

Version 2.14.2 from 12/12/2017

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

Version 2.14.3 dated 12/17/2017

Added a clarification on the restriction on the maximum number of the subject of calculation .

Version 2.14.4 from 02/13/2018

Fixed an example of a signature distorted by the Word .

Version 2.15.0 from 02/27/2018

Added the ability to transfer tags 1225, Supplier's name, 1226, Supplier's TIN, 1171, Supplier's phone number associated with the subject of calculation.

Version 2.16.0 from 05/15/2018

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

Clarifications have been made regarding the length of 1084, Optional User Attribute.

Clarifications regarding the maximum length of some parameters.

Version 2.16.1 from 05/24/2018

The restriction on the maximum number of positions is described.

Version 2.17.0 from 06/19/2018

Added transfer of tags sign of the agent on the subject of calculation 1222, agent data 1223, unit of measurement of the subject of calculation 1197, additional attribute of the subject of calculation 1191.

Version 2.18.0 from 08/01/2018

Added the transfer of the tag additional details of the check, 1192.

Changed signature example to more correct with normal quotation marks .

Version 2.19.0 from 08/03/2018

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

- ✓ Buyer (client), 1227
- ✓ TIN of the buyer (client), 1228
- ✓ Code of the country of origin of goods, 1230
- ✓ Customs declaration number, 1231
- ✓ Excise, 1229

Version 2.20.0 from 08/31/2018

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

Version 2.20.1 from 10/23/2018

Amendments and clarifications have been made to the protocol.

Version 2.21.0 from 12/01/2018

Changed descriptions of tags 1139, 1141, 1151, 1153, 1162.

Version 2.22.0 from 04/28/2019

Added values for tag 1212 and 1030.

Version 2.22.1 from 06/11/2019

Added link to check.

Version 2.23.0 from 06/15/2019

callbackUrl parameter has been added to the receipt creation and correction receipt requests .

Version 2.24.0 from 10/10/2019

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

Version 2.24.1 from 11/25/2019

The IP address of the test environment has been changed.

Version 2.25.0 from 11/26/2019

The ability to transfer tags 1021 Cashier, 1117 E-mail address of the check sender, 1203 TIN of the cashier has been added to the check.

Version 2.26.0 from 12/09/2019

Added a method for requesting the status of a cash register in the group p.2.5.

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

Version 2.27.0 from 02/27/2020

Added new values of the attribute of the subject of calculation, tag 1212

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

Version 2.27.1 from 08/03/2020

Added note about differences in the numbering of tax rates in the fields **vat1Sum, vat2Sum, vat3Sum, vat4Sum, vat5Sum, vat6Sum** and in the tax field (tag 1199).

Added remark about rounding in the **totalSum field** (tag 1020)

Valid values for the **paymentSubjectType field have been changed** (tag 1212). Since version 2.27.0, values from 1 to 26 are allowed.

Version 2.27.2 from 02/05/2021

Added introductory information about cloud cash desks and description of basic terms.

Added detailed information about connection parameters for the production environment and two test environments.

Added detailed information about files to test in Git .

Added note about Postman application and SignatureGenerator.exe.

Updated check limit information. 32768 bytes accounted for the size of a binary fiscal document. A JSON document can contain about 250 positions or 100 KB.

Version 2.28.0 from 08/13/2021

ignoreItemCodeCheck flag has been added to the body of the request to create a document .

FFD 1.2 tags have been added to the request to create a document.

Added requests for the formation of the FFD correction check 1.2 p.2.7 and p.2.8.

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

Version 2.28.1 from 01/19/2022

Added parameter Sign of the subject of calculation, 1212

Added parameter Planned status, 2003

Version 2.28.2 from 02/26/2022

Document correction

Version 2.28.3 from 25/03/2023

The **callbackApiKey** parameter has been added to the body of the request to create a document.

Version 2.29.0 from 20/07/2023

The IP of the production environment has been updated. New IP 81.27.253.4

Old IP 62.76.112.48

Version 2.29.1 from 15/11/2023

Updated data in the fields senderEmail – tag 1117, customerContact – tag 1008

Added the ability to send the value "none"

Version 2.30.0 from 05/12/2024

Added support for VAT rates 5%, 7%, 5/105 и 7/107.

Attribute tax 7, tax 8, tax 9, tax 10;

Added attributes tax7Sum, tax8Sum, tax9Sum, tax10Sum. The amount of VAT in the correction receipt;

Added attributes vat7Sum, vat8Sum, vat9Sum, vat10Sum. The amount of VAT in the receipt / receipt of the FFD correction 1.2;

Attribute validation has been removed paymentSubjectType 15, 16.