The interaction protocol with Checkbox

Version 2.15.0 from 27.02.2018

Contents

[2.. Description 3](#_Toc512501783)

[Example of message with Signature 3](#_Toc512501784)

[2. Requests 6](#_Toc512501785)

[2.1 Check creation 6](#_Toc512501786)

[2.1.1 Request body 6](#_Toc512501787)

[2.1.1.1 Document content 6](#_Toc512501788)

[2.1.1.2 Payment subject 7](#_Toc512501789)

[2.1.1.3 Check close parameters 8](#_Toc512501790)

[2.1.1.4 Payment 8](#_Toc512501791)

[2.1.1.5 Additional user’s attribute 8](#_Toc512501792)

[2.1.1.6 Supplier’s data 8](#_Toc512501793)

[2.1.2 Response body with request processing errors 10](#_Toc512501794)

[2.2 Check status 12](#_Toc512501795)

[2.2.1 Response body 12](#_Toc512501796)

[2.3 Check of correction creation 14](#_Toc512501797)

[2.3.1 Request body 14](#_Toc512501798)

[2.3.1.1 Document content 14](#_Toc512501799)

[2.3.2 Response body with request processing errors 15](#_Toc512501800)

[2.4 Check correction status 17](#_Toc512501801)

[2.4.1 Response body 17](#_Toc512501802)

[3. Swagger 19](#_Toc512501803)

# Description

Interaction with internet-checkbox is carried-out by messaging with web-service.

Transport – https with use of client and server certificates for mutual authentication.

Message format – JSON.

Messages are protected from any effects and changes by encryption with SHA256-RSA key, no less 2048 Kbit, PKCS1 padding.

The signature is transferred in http-heading X-Signature and is encrypted by base64 code.

Internet-checkbox has public part of client key and verifies received signature.

For signature generation, the message converted into bytes with UTF-8 encoding is used.

Internet-checkbox generates fiscal documents according Format of Fiscal Data (**FFD 1.05**). Documents are generated by Fiscal Store (FN), this fact imposes restrictions on coding of the transferred lines. Messaging with cashbox using http is carried out with UTF-8 encoding, but at recording in the FN lines are transferred with **CP866** encoding that maintains less symbols amount than UTF-8. For instance, CP866 has no «» symbols, so

FN will record *Juice Pear* instead of *Juice «Pear»*.

Also, Checkbox does not provide full format-logistic control. For example, buyer’s Phone-number or e-mail (tag 1008) according FFD has format +{F} или {S}@{S}, but internet checkbox allows any line from 1 to 64 symbols. Internet checkbox does not control existence/absence of any agent tags, so we highly recommend to learn original FFD <https://www.nalog.ru/html/sites/www.new.nalog.ru/doc/pril2_fns229_210317.docx> and to watch closely for formats, existence/absence of tags depending on your concrete work scenario.

In addition, it is worth to pay attention at unreliability of network and, as a result, the failures connected with network interaction. In case of unsuccessful sending of the document it is necessary to execute several repetitions with an acceptable timeout, for example, 5 attempts with an interval of 10-20 seconds (figures are provided for descriptive reasons and can differ depending on a concrete situation).

### Example of message with Signature

POST /api/v2/documents HTTP/1.1

Content-Length: 363

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

X-Signature: K/YpoNDWcG5FOrq4VZgBKVK7IyvYQnqJ1YLUUe5N3dTnGVzaGV6s3HLh0HcOFrp54eauMdU2O3SPRDJJBkpzhFF/Ydf7FpfAb9Tj1XpRoYt3Q9acciTFferE8gb5niW/yq3bmYJQiSqM8hSIDhoqF4HcaWKHiXPsu/XusgjsS1RqQeC8eH96/m2e35cUcAK980Kv5dCj6ePByC+9Q4h8g1QqSpNQdG0UsayFo2o5yCWMe1fbLE0EseRUUSZ2Z79YS42H6S+psozd/WNTtCkF6p3PvabVmOMBvZaPPtij//51W1cS7bm9civYH/FgjpxVm476F4com+zhutXhe2VHlg==

{"Id": "34658763458763458","INN": "3123011520","Content": {"Type": 1,"Positions": [{"Quantity": 1.000,"Price": 123.45,"Tax": 6,"Text": "Bread"},{"Quantity": 2.000,"Price": 4.45,"Tax": 4,"Text": "Lighter"}],"CheckClose": {"Payments": [{"Type": 1,"Amount": 123.45},{"Type": 2,"Amount": 8.90000}],"TaxationSystem": 1},"CustomerContact": "foo@example.com"}}

RSA-key:

<RSAKeyValue>

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

  <Exponent>AQAB</Exponent>

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

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

  <DP>Fo5KiNCJCtCbpFfH4XVM5UJdXPXTbNBHBdlYMJ9AddTl5Ijrt50ExgLFu4oMPMsYXryS61LI2WT5XCqIvmbcnhYbambgWLOKYuZUUYSr2kS67So5FUCunWaGhTdx2bRLQVqwm6kiXDPDnMRAViiCHXWqk/VsrXheVymhLqNK440=</DP>

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

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

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

</RSAKeyValue>

Example of creating Signature using PHP using <https://github.com/phpseclib/phpseclib>

function computeSignature($data)

{

    $rsa = new RSA();

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

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

    $rsa->setHash(‘sha256’);

    $rsa->setMGFHash(‘sha256’);

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

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

}

Example of creating Signature using PHP with Openssl. For this method a key in PEM format is needed. XML key can be converted using any XML to PEM convertors, for example, <https://superdry.apphb.com/tools/online-rsa-key-converter>:

function computeSignature($data)

{

    $key = <<<KEYDATA

    -----BEGIN RSA PRIVATE KEY-----

    MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQC3ycL8S2HxRptB

    te7yl2uje/s2pRqdXxj6D3ZiBPvPXGqQEtEddnWC6aXc/GuqM1f0C86a7xH6poo7

    Id8lbQ9xEMvMKghRwc0DCkM78TmPpYBosi/uACNO3Kv2QkH2t8lqlqtWIk1m7dFJ

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

    ocFFonRd0TeWHY54T384XQG0vCJg8MqxVPEh6Rs1/uX8NETL5htQ7Fatx54deu9t

    guIZZ5w/RrsKocaP1k1jWglOErcDCtJ3jIdr1afH8ZplQ21a53Ufo/2DexVf6xFX

    3G2cj3p3AgMBAAECggEAPUfM+Aq6kZSVWAetsL3EajKAxOuwQCDhVx+ovW4j+DQ8

    Y+WiTEyfShNV9qVD0PBltz3omch1GjpFhQn6OaRvraeIDH9HXttb3Fojr2zzYG4y

    rrYbPSRWoYj63ZwiIP2O7zdl0caGQHezfNcYa2N0NTG99DGc3/q6EnhlvjWQsSbi

    EjmxcPx8fmV1i4DoflMQ383nsixAFapgrROUAtCgMvhWn1kSeoojKd+e4eKZxa/S

    NYulsBJWNFkmo1CZH4YtqlPM+IwYeDUOnOUGNxGurRZ3qQdWs2N2ZQhnrvlh+zpz

    urD2hwAz6gQXP7mxxMR1xHtAD8XQ+w4OiJK6VWjoIQKBgQDdZJvvZrV6tvqNwuTJ

    kDZjbVU0iKkbP61rVE/6JpyzfGeS0WzGBNiCpbK3pJZnatK2nS7i9v8gAfIqGAk8

    1NRKLa7Qbjgw6xHEwL8VZMXzN3KsMXgGM8EziPzicCYT8Vbi/kXyV0OrqRz3rMQ+

    JOTkWRrcw943yYyTr84Dn0l0XQKBgQDUhFWJ3lKwOs7AlAAQqR1PjfpcRvSxVZ70

    BxTwnJoIQQyPQ0/OjCc1sit5s+h8xh0MeKSilCmvZerFlgNtvsCd6geSERXbpN+k

    9Vs3jAEkVeKHeUA/afmGqGCocanlarYu7uNRLfvpG7DduHBb4yJale/XGExNnwC0

    N+dkUU284wKBgBaOSojQiQrQm6RXx+F1TOVCXVz102zQRwXZWDCfQHXU5eSCa7ed

    BMYCxbuKDDzLGF68kutSyNlk+VwqiL5m3J4WG2pm4FizimLmVFGEq9pEuu0qORVA

    rp1mhoU3cdm0S0FasJupIlwzw5zEQFYogh11qpP1bK14XlcpoS6jSuONAoGBAJqM

    EljM4X1fhvPtrY5wLeyo56UrxM8h4RK+A7Bncm0GQUf+P4+JxQn7pDpBZ5U1zfI/

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

    P5LnyiDgDmNvueZY66bM2zqlZPgd5bkp3pDJv6rZAoGAaP5e5F1j6s82Pm7dCpH3

    mRZWnfZIKqoNQIq2BO8vA9/WrdFI2C27uNhxCp2ZDMulRdBZcoeHcwJjnyDzg4I4

    gBZ2nSKkVdlN1REoTjLBBdlHi8XkiXzxvpItc2wjNC2AKHaJqj/dnh3bbTAQD1iU

    AxPmmLJYYkhfZ2i1IrTVxZE =

    -----END RSA PRIVATE KEY-----

    KEYDATA;

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

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

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

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

    return base64\_encode($res);

}

Example of messaging using 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));

Example of creating Signature using C# using standard .net core libraries:

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));

    }

}

Example of creating Signature using C# using standard .net framework full libraries:

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”));

    }

}

# 2. Requests

## 2.1 Check creation

Check creation is asynchronous request, after its performance check is placed in queue for processing.

Except fiscal data client should send unique ID, that is used to get information about check status (p. 2.2).

Also, this ID provides idempotence of requests. This ID should be unique within the organization. Taxpayer identification number (INN) is used for determination of checkbox that can be used for formation of the check.

Tag “Group” in the request can be null or not stated, in that case, check will be send in default group.

Default group is stated group “Main”.

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

### 2.1.1 Request body

|  |  |  |
| --- | --- | --- |
| id | Document identifier | Line from 1 to 64 symbols |
| Inn | Taxpayer identification number of organization that is forming check | Line 10 or 12 symbols |
| group | Group of checkboxes, used for check formation | Line from 1 to 32 symbols or null |
| content | Document content | Structure p.2.1.1.1 |
| key | Name of the key that is used for signature verification. Optional parameter. If the name of the key is not stated, default key will be used for signature verification. | Line from 1 to 32 symbols or null |

### 2.1.1.1 Document content

|  |  |  |
| --- | --- | --- |
| type | Attribute (type) of payment, 1054:   1. Income 2. Income return 3. Expenditure 4. Expenditure return | Number |
| positions | List of payment subjects, 1059 | Massif of structures  p.2.1.1.2 |
| checkClose | Check close parameter | Structure p.2.1.1.3 |
| customerContact | Buyer’s phone-number or e-mail, 1008 | Line from 1 to 64 symbols, format +{F} or {S}@{S} |
| agentType | Agent attribute (type), 1057. Bit field, where bit number means that organization  rendering service to the buyer (client) is:  0 – bank payment agent  1 – bank payment subagent  2 – payment agent  3 – payment subagent  4 – attorney  5 – commission agent  6 – other agent  *Check can include requisites «Agent attribute (type)» (tag 1057), only if registration report and (or) current re-registration report includes requisite «Agent attribute (type)» (tag 1057) identical to value of the requisite «Agent attribute (type)» at check.* | Number from 1 to 127, not obligatory field |
| paymentTransferOperatorPhoneNumbers | Payment transfer operator’s phone number, 1075 | Massif of lines from 1 to 19 symbols, format +{F}, not obligatory field |
| paymentAgentOperation | Payment agent’s operation, 1044 | Line from 1 to 24 symbols, not obligatory field |
| paymentAgentPhoneNumbers | Payment agent’s phone number, 1073 | Massif of lines from 1 to 19 symbols, format +{F}, not obligatory field |
| paymentOperatorPhoneNumbers | Payment operator’s phone number, 1074 | Massif of lines from 1 to 19 symbols, format +{F}, not obligatory field |
| paymentOperatorName | Payment transfer operator’s name, 1026 | Line from 1 to 64 symbols, not obligatory field |
| paymentOperatorAddress | Payment transfer operator’s address, 1005 | Line from 1 to 244 symbols, not obligatory field |
| paymentOperatorINN | Payment transfer operator’s INN, 1016 | Line from 10 to 12 symbols, format FFFFFFFFFF, not obligatory field |
| supplierPhoneNumbers | Supplier’s phone number, 1171 | Massif of lines from 1 to 19 symbols, format +{F}, not obligatory field |
| additionalUserAttribute | Additional user’s attribute, 1084 | Structure p.2.1.1.5, or null |

### 2.1.1.2 Payment subject

|  |  |  |
| --- | --- | --- |
| quantity | Amount of payment subjects, 1023 | Decimal number up to 6 decimal places \* |
| price | Price for a unit of payment subject including discount and margins, 1079 | Decimal number up to 2 decimal places \* |
| tax | Rate value added tax (VAT), 1199:  1 – VAT rate 18%  2 – VAT rate 10%  3 – VAT rate calc. 18/118  4 – VAT rate calc. 10/110  5 – VAT rate 0%  6 – VAT isn't assessed | Number from 1 to 6 |
| text | Name of payment subject, 1030 | Line up to 128 symbols |
| paymentMethodType | Payment method attribute (type), 1214:  1 – Advance payment 100%  2 – Partial advance payment  3 – Advance  4 – Full payment  5 – Partial payment or loan  6 – Transfer on credit  7 – Payment of the credit | Number from 1 to 7 or null. If “null”, 4 (full payment) states by default. |
| paymentSubjectType | Payment subject attribute (type), 1212:  1 – Commodity  2 – Excise commodity  3 – Work  4 – Service  5 – Gambling rate  6 – Gambling prize  7 – Lottery ticket  8 – Lottery prize  9 – Granting results of intellectual activity  10 – Payment  11 – Agency fee  12 – Compound payment subject  13 – Other payment subject | Number from 1 to 13 or null. If “null’, 1 (commodity) states by default. |
| nomenclatureCode | Nomenclature code, 1162 | Line containing base 64 coded massif from 1 to 32 bytes or null |
| supplierInfo | Supplier’s info, 1224 | Structure p.2.1.1.6, or null |
| supplierINN | Supplier’s INN, 1226 | Line from 10 to 12 symbols, format FFFFFFFFFF, not obligatory field |

### 2.1.1.3 Check close parameters

|  |  |  |
| --- | --- | --- |
| payments | Payments | Massif of structures p.2.1.1.4 |
| taxationSystem | Taxation system, 1055:  0 – General taxation system, GTS (ОСН)  1 – Simplified taxation system (income), STS income (УСН 6%)  2 – Simplified taxation system (income-expenditures), STS income-expenditures (УСН 15%)  3 – The single tax on imputed income (ЕНВД)  4 – Unified agricultural tax (ЕСН)  5 – Patent taxation system (Патент) | Number from 0 to 5 |

### 2.1.1.4 Payment

|  |  |  |
| --- | --- | --- |
| type | Payment type:  1 – amount in check by cash, 1031  2 – amount in check by electronic payment instrument, 1081  14 – amount in check by advance payment (offset of advance or previous payments), 1215  15 – amount in check by post-payment (on credit), 1216  16 – amount in check by counter granting, 1217 | Number from 1 to 16 |
| amount | Payment amount | Decimal number up to 2 decimal places \* |

\* Maximal payment amount in check 99 999 999,99 rubles. Maximal quantity amount \* price after rounding also 99 999 999.99. Maximal price per payment subject unit 99 999 999.99.

### 2.1.1.5 Additional user’s attribute

|  |  |  |
| --- | --- | --- |
| name | Name of additional user’s attribute, 1085 | Line from 1 to 64 symbols |
| value | Value of additional user’s attribute, 1086 | Line from 1 to 175 symbols |

### 2.1.1.6 Supplier’s data

|  |  |  |
| --- | --- | --- |
| phoneNumbers | Supplier’s phone number, 1171 | Massif of lines from 1 to 19 symbols, format +{F}, not obligatory field |
| name | Supplier’s name, 1225 | Line up to 240 symbols.  Attention: these 240 symbols include supplier’s phone number +4 symbols for every phone number. For instance, if 2 phone numbers were transferred with 12 and 14 symbols each, then maximal length of supplier’s name is 240 – (12 + 4) – (14 + 4) = 206 symbols |

As a response, api response following http status-codes:

* 201 Created – Check is created and added in queue for processing, empty response body
* 401 Unauthorized – client certificate has failed to provide valid authentication credentials
* 409 Conflict – check with this ID already exists in the system, empty response body
* 400 Bad Request – transferred data includes validation errors, or signature has failed validation, response body p. 2.1.2
* 503, Service Unavailable – queue is overcrowded, hidder Retry-After with timeout in seconds is responded. Repeat request after this time. Response body p.2.1.2.

### 2.1.2 Response body with request processing errors

|  |  |  |
| --- | --- | --- |
| errors | Massif of request processing errors | Massif of lines |

Request example:

{

  "id": "12345678990",

  "inn": "123456789012",

  "group": "Main",

  "content": {

    "type": 1,

    "positions": [

      {

        "quantity": 1.000,

        "price": 123.45,

        "tax": 6,

        "text": "Bread",

        "paymentMethodType": 4,

        "paymentSubjectType": 1

      },

      {

        "quantity": 2.000,

        "price": 4.45,

        "tax": 4,

        "text": "Lighter",

"paymentMethodType": 3,

        "paymentSubjectType": 13

      }

    ],

    "checkClose": {

      "payments": [

        {

          "type": 1,

          "amount": 123.45

        },

        {

          "type": 2,

          "amount": 8.90000

        }

      ],

      "taxationSystem": 1

    },

    "customerContact": "foo@example.com"

  }

}

Request example with agent data, additional user’s and supplier’s requisites:

{

  "id": "12345678990",

  "inn": "123456789012",

  "group": "Main",

  "key": "1234567",

  "content": {

    "type": 1,

    "positions": [

      {

        "quantity": 1.000,

        "price": 123.45,

        "tax": 6,

        "text": "Bread",

        "paymentMethodType": 4,

        "paymentSubjectType": 1,

        "nomenclatureCode": "igQVAAADMTIzNDU2Nzg5MDEyMwAAAAAAAQ==",

        "supplierInfo": null

      },

      {

        "quantity": 2.000,

        "price": 4.45,

        "tax": 4,

        "text": "Lighter",

        "paymentMethodType": 3,

        "paymentSubjectType": 13,

        "supplierINN": "9715225506",

        "supplierInfo": {

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

          "name": "ПАО \"Адамас\""

        }

      }

    ],

    "checkClose": {

      "payments": [

        {

          "type": 1,

          "amount": 123.45

        },

        {

          "type": 2,

          "amount": 8.90000

        }

      ],

      "taxationSystem": 1

    },

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

    "agentType": 127,

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

    "paymentAgentOperation": "Какая-то операция",

    "paymentAgentPhoneNumbers": [ "+79260000003" ],

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

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

    "paymentOperatorAddress": "Москва, Мастеркова 4",

    "paymentOperatorINN": "9715225506",

    "supplierPhoneNumbers": [ "+74957870004" ],

    "additionalUserAttribute": {

      "name": "Favorite quote",

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

    }

  }

}

Response example with errors:

{

  "errors": [

    "Не указан идентификатор документа 'Id'",

    "Не указан ИНН организации 'INN'",

    "Отсутствует содержимое документа 'Content'"

  ]

}

{

  "errors": [

    "Document identifier is not stated 'Id'",

    "INN of organization is not stated 'INN'",

    "Field is empty 'Content'"

  ]

}

## 2.2 Check status

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

**{inn}** – Taxpayer identification number (INN) of organization that is forming check

**{document\_id}** – Document identifier stated at its creation

In this request SHA256-RSA key is not used.

As a response: api response following status-codes:

* 202 Accepted – Check is created and added in queue for processing, but not created yet, empty response body
* 400 Bad Request – organization not found, request with this ID not found
* 401 Unauthorized – client certificate has failed to provide valid authentication credentials
* 200 OK – check is created, response body p.2.2.1.

### 2.2.1 Response body

|  |  |  |
| --- | --- | --- |
| id | Document identifier | Lines from 1 to 64 symbols |
| deviceSN | Serial Number of device created check | Line up to 20 symbols |
| deviceRN | Register Number of device created check | Line up to 20 symbols |
| fsNumber | Fiscal storage (FN) number | Line 16 symbols |
| ofdName | Name of OFD | Line up to 256 symbols |
| ofdWebsite | Web-site of OFD | Line up to 58? symbols |
| ofdINN | INN of OFD | Line up to 12 symbols |
| fnsWebsite | Web-site of Federal Tax Service | Line up to 256 symbols |
| companyINN | Organization INN | Line 12 symbols |
| companyName | Organization name | Line up to 256 symbols |
| documentNumber | Fiscal document (check) number | Number |
| shiftNumber | Shift number | Number |
| documentIndex | Fiscal document (check) number per shift | Number |
| processedAt | Time and date when fiscal document was created in FN | Time in format ISO8601 |
| content | Content of check | Structure p.2.1.1.1 |
| change | Change | Decimal number up to 2 decimal places |
| fp | Fiscal sign | Line 10 symbols |

Response example:

{

  "id": "12345678990",

  "deviceSN": "0000000000001358",

  "deviceRN": "0000000400054952",

  "fsNumber": "9999078900001341",

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

  "ofdWebsite": "www.ofd-ya.ru",

  "ofdinn": "7728699517",

  "fnsWebsite": "www.nalog.ru",

  "companyINN": "123456789012",

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

  "documentNumber": 117,

  "shiftNumber": 20,

  "documentIndex": 5,

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

  "content": {

    "type": 1,

    "positions": [

      {

        "quantity": 1.000,

        "price": 123.45,

        "tax": 6,

        "text": "Bread",

        "paymentMethodType": 4,

        "paymentSubjectType": 1

      },

      {

        "quantity": 2.000,

        "price": 4.45,

        "tax": 4,

        "text": "Lighter",

        "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"

}

## 2.3 Check of correction creation

Check of correction creation is asynchronous request, after its performance check is placed in queue for processing.

Except fiscal data client should send unique ID, that is used to get information about check of correction status (p. 2.4).

Also, this ID provides idempotence of requests. This ID should be unique within the organization. Taxpayer identification number (INN) is used for determination of checkbox that can be used for formation of the check.

Tag “Group” in the request can be null or not stated, in that case, check will be send in default group.

Default group is stated group “Main”.

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

### 2.3.1 Request body

|  |  |  |
| --- | --- | --- |
| id | Document identifier | Line from 1 to 64 symbols |
| Inn | Taxpayer identification number of organization that is forming check | Line 10 or 12 symbols |
| group | Group of checkboxes, used for check formation | Line from 1 to 32 symbols or null |
| content | Document content | Structure p.2.1.1.1 |
| key | Name of the key that is used for signature verification. Optional parameter. If the name of the key is not stated, default key will be used for signature verification. | Line from 1 to 32 symbols or null |

### 2.3.1.1 Document content

|  |  |  |
| --- | --- | --- |
| correctionType | Correction type, 1173:  0. Itself  1. By warrant | Number |
| type | Attribute (type) of payment, 1054:  1. Income  3. Expenditure | Number |
| description | Correction description, 1177 | Line from 1 to 244 symbols |
| causeDocumentDate | Date of correction cause document, 1178.  This tag must be always transferred with time 00:00:00 | Time in format ISO8601 |
| causeDocumentNumber | Number of correction cause document, 1179 | Line from 1 to 32 symbols |
| totalSum | Total amount in check, 1020  Сумма расчета, указанного в чеке (БСО), 1020 | Decimal number up to 2 decimal places |
| cashSum | Amount in check by cash, 1031 | Decimal number up to 2 decimal places |
| eCashSum | Amount in check by electronic payment instrument, 1081 | Decimal number up to 2 decimal places |
| prepaymentSum | Amount in check by advance payment (offset of advance or previous payments), 1215 | Decimal number up to 2 decimal places |
| postpaymentSum | Amount in check by post-payment (on credit), 1216 | Decimal number up to 2 decimal places |
| otherPaymentTypeSum | Amount in check by counter granting, 1217 | Decimal number up to 2 decimal places |
| tax1Sum | Amount of VAT in check at rate 18%, 1102 | Decimal number up to 2 decimal places |
| tax2Sum | Amount of VAT in check at rate 10%, 1103 | Decimal number up to 2 decimal places |
| tax3Sum | Amount of VAT in check at rate 0%, 1104 | Decimal number up to 2 decimal places |
| tax4Sum | Payment amount where VAT isn't assessed, 1105 | Decimal number up to 2 decimal places |
| tax5Sum | Amount of VAT in check at calc. rate 18/118, 1106 | Decimal number up to 2 decimal places |
| tax6Sum | Amount of VAT in check at calc. rate 10/110, 1107 | Decimal number up to 2 decimal places |
| taxationSystem | Taxation system, 1055:  0 – General taxation system, GTS (ОСН)  1 – Simplified taxation system (income), STS income (УСН 6%)  2 – Simplified taxation system (income-expenditures), STS income-expenditures (УСН 15%)  3 – The single tax on imputed income (ЕНВД)  4 – Unified agricultural tax (ЕСН)  5 – Patent taxation system (Патент) | Number |

As a response, api response following http status-codes:

* 201 Created – Check is created and added in queue for processing, empty response body
* 401 Unauthorized – client certificate has failed to provide valid authentication credentials
* 409 Conflict – check with this ID already exists in the system, empty response body
* 400 Bad Request – transferred data includes validation errors, or signature has failed validation, response body p. 2.3.2
* 503, Service Unavailable – queue is overcrowded, hidder Retry-After with timeout in seconds is responded. Repeat request after this time. Response body p.2.3.2.

### 2.3.2 Response body with request processing errors

|  |  |  |
| --- | --- | --- |
| errors | Massif of request processing errors | Massif of lines |

Request example:

{

  "id": "12345678990",

  "inn": "123456789012",

  "group": "Main",

  "content": {

    "correctionType": 1,

    "type": 1,

    "description": "You don’t want, but have to",

    "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

  }

}

Response example with errors:

{

  "errors": [

    "Не указан идентификатор документа 'Id'",

    "Не указан ИНН организации 'INN'",

    "Отсутствует содержимое документа 'Content'"

  ]

}

{

  "errors": [

    "Document identifier is not stated 'Id'",

    "INN of organization is not stated 'INN'",

    "Field is empty 'Content'"

## 2.4 Check correction status

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

**{inn}** – Taxpayer identification number (INN) of organization that is forming check

**{document\_id}** – Document identifier stated at its creation

In this request SHA256-RSA key is not used.

As a response: api response following status-codes:

* 202 Accepted – Check is created and added in queue for processing, but not created yet, empty response body
* 400 Bad Request – organization not found, request with this ID not found
* 401 Unauthorized – client certificate has failed to provide valid authentication credentials
* 200 OK – check is created, response body p.2.4.1.

### 2.4.1 Response body

|  |  |  |
| --- | --- | --- |
| id | Document identifier | Lines from 1 to 64 symbols |
| deviceSN | Serial Number of device created check | Line up to 20 symbols |
| deviceRN | Register Number of device created check | Line up to 20 symbols |
| fsNumber | Fiscal storage (FN) number | Line 16 symbols |
| ofdName | Name of OFD | Line up to 256 symbols |
| ofdWebsite | Web-site of OFD | Line up to 58? symbols |
| ofdINN | INN of OFD | Line up to 12 symbols |
| fnsWebsite | Web-site of Federal Tax Service | Line up to 256 symbols |
| companyINN | Organization INN | Line 12 symbols |
| companyName | Organization name | Line up to 256 symbols |
| documentNumber | Fiscal document (check) number | Number |
| shiftNumber | Shift number | Number |
| documentIndex | Fiscal document (check) number per shift | Number |
| processedAt | Time and date when fiscal document was created in FN | Time in format ISO8601 |
| content | Content of check | Structure p.2.1.1.1 |
| fp | Change | Decimal number up to 2 decimal places |

Response example:

{

  "id": "12345678990",

  "deviceSN": "0000000000001358",

  "deviceRN": "0000000400054952",

  "fsNumber": "9999078900001341",

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

  "ofdWebsite": "www.ofd-ya.ru",

  "ofdinn": "7728699517",

  "fnsWebsite": "www.nalog.ru",

  "companyINN": "123456789012",

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

  "documentNumber": 117,

  "shiftNumber": 20,

  "documentIndex": 5,

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

  "content": {

    "correctionType": 1,

    "type": 1,

    "description": "You don’t want, but have to",

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

    "causeDocumentNumber": "ФЗ-54",

    "totalSum": 17.25,

    "cashSum": 1.23,

    "eCashSum": 2.34,

    "prepaymentSum": 5.67,

    "postpaymentSum": 4.56,

    "otherPaymentTypeSum": 3.45,

    "tax1Sum": 1.34,

    "tax2Sum": 2.34,

    "tax3Sum": 3.34,

    "tax4Sum": 4.34,

    "tax5Sum": 5.34,

    "tax6Sum": 6.34,

    "taxationSystem": 1

  },

  "change": 974.01,

  "fp": "2364009522"

}

# 3. Swagger

Description api well in a look OpenAPI Specification(OAS).

By url **/swagger** swagger-ui is accessible.

By url **/swagger/v2/swagger.json** – specifications.