



Reģionālā
Investīciju Banka



RIB Payment Gateway API

Changelog	3
General information	4
Introduction	4
Service basics and testing instructions	4
Certificate generation	5
Transport authentication certificate	5
Signature key pair creation	6
Authentication and security	7
Mutual TLS authentication	7
HTTP signature	7
Account information services	11
Past days statement	11
Payment file import	15
Payment initiation	15
Payment status	18
Payment cancellation	19

Date	Change
04.04.2025.	Initial version of the document
03.06.2025.	The instructions for creating certificates have been clarified
14.07.2025.	Clarified "Payment cancellation" description

Introduction

RIB Payment Gateway API (Application Programming Interface) is a solution for direct connection and data exchange between the bank and relevant software of the client. It offers a possibility to get account information, as well as forwarding of payment orders round-the-clock. This helps you to use banking services in secure, convenient, and automated way.

Available Services:

- Account information for RIB accounts: past days statement (ISO XML camt.052)
- Payment initiation (ISO XML pain.001) and Payment status (ISO XML pain.002)
- Payment file (pain.001) import to RIB internet bank for confirmation

API version number is specified in URL path:
<https://pg.ribbank.com:8080/v1/>

Documentation for new releases will be published at least 3 months prior unless RIB is forced to terminate or make changes sooner due to regulatory, compliance, security or customer support reasons.

Clients will be notified if the service offering is terminated, or changes have direct impact for existing usage.

Service basics and testing instructions

APIs are based on REST architecture; data transfer is ensured using:

- HTTPS (HTTP 1.1 and TLS 1.2) protocol;
- PKI-based client authentication with RIB issued certificate;
- ISO 20022 XML format for standard account information and payment initiation services – format descriptions aren't part of RIB Gateway service specification but added as separate documents;
- JSON format support for custom services and endpoints;
- UTF-8 encoding is used by default for all endpoints;
- Domain to access PRODUCTION services:
<https://pg.ribbank.com:8080/v1/> (IP whitelisting is needed);
- Domain for TEST services is available on customer request. If you need test environment, please contact RIB.

When deciding to start using the described services, you should begin with reviewing the provided documentation.

It's recommended to initially try out API-s in test environment and making sure that everything is working for you as expected. It's required to follow general best practices of API usage.

Certificate generation

Transport public certificate

To create connection with API Payment Gateway, transport authentication certificate is necessary. In order to receive it, you will have to generate private key and certificate signing request (CSR). Private key shall remain to you, however, CSR shall be sent to bank.

To create encrypted private key, use these commands:

```
openssl genrsa -aes256 -out private.key 2048
```

Option to create password for file will be provided to you. **In order to use private key without password**, use following command:

```
openssl genrsa -out private.key 2048
```

To create certificate signing request (CSR) use following command:

```
openssl req -new -sha256 -key privat.key -out mycompany.csr
```

Enter all requested information (for example):

```
Country Name (2 letter code) [XX]: LV
State or Province Name (full name) []: Riga
Locality Name (eg, city) [Default City]: Riga
Organization Name (eg, company) [Default Company Ltd]: My Company
Name
Organizational Unit Name (eg, section) []: IT
Common Name (eg, your name or your server's hostname) []: RIB
Payment Gateway agreement number*
Email Address []: mycompany@mycompany.com
```

Remaining fields can be left empty (press Enter to continue).

Please **enter the following 'extra' attributes** to be sent with your certificate request:

```
A challenge password []:
An optional company name []:
```

Change the file extension from .csr to .txt and send the file named mycompany.txt to the Bank. In return, you will receive a transport certificate required to establish a connection with the RIB Payment Gateway API.

***RIB Payment Gateway agreement number** - Your RIB Payment Gateway contract number (e.g., JP/PG-123456).

Certificate generation

Signature key pair creation

In order to initiate payments, receive account statement and execute other requests, you will need to generate private and public key pair.

Execute following command **to create encrypted private key**:

```
openssl genpkey -algorithm RSA -aes256 -out private_sign.key -pkeyopt  
rsa_keygen_bits:2048
```

Option to set password for file access will be offered to you. **To create private key without password**, execute following command:

```
openssl genpkey -algorithm RSA -out private_sign.key -pkeyopt  
rsa_keygen_bits:2048
```

Create certificate signing request executing following command:

```
openssl req -new -key private_sign.key -out mycompany_sign.csr
```

Enter all required data (example):

```
Country Name (2 letter code) [XX]: LV  
State or Province Name (full name) []: Riga  
Locality Name (eg, city) [Default City]: Riga  
Organization Name (eg, company) [Default Company Ltd]: My Company  
Name  
Organizational Unit Name (eg, section) []: IT  
Common Name (eg, your name or your server's hostname) []: User code*  
Email Address []: mycompany@mycompany.com
```

Other fields can be left empty (press Enter).

Please **enter the following 'extra' attributes** to be sent with your certificate request:

```
A challenge password []:  
An optional company name []:
```

Create public certificate:

```
openssl x509 -req -in mycompany_sign.csr -signkey private_sign.key -out  
mycompany_public_sign_usercode.crt -days 730 -sha512
```

Change the file extension from .crt to .txt and send the file named mycompany_public_sign_usercode.txt to the Bank.

***User code** - The user ID is specified in your RIB Payment Gateway agreement.

Mutual TLS authentication

API communication uses certificate-based authentication (X.509 public key infrastructure (PKI) standard) over HTTPS (HTTP/1.1 over mutual TLS1.2). Certificate for authentication is issued by RIB, based on Certificate Signing Request (CSR) provided by the client. Certificate validity period is 3 years and new certificate needs to be ordered by the client from RIB before the expiry date.

For instructions on how to generate CRS, please refer to section **Certificate generation** in this document.

HTTP signature

For all requests it's mandatory to include HTTP signature in message header to ensure that the data sent to bank has not been tampered with during or after the transit. Signature is created with private key you provided for RIB Gateway authentication. RIB validates the signature using public key you provided for RIB Gateway and request is only processed when signature is present and valid.

For signature generation you must use private key from signature certificate you provided to Bank.

Creating signature

Payload for creating signature contains three parts – URI of the request to be performed, unix timestamp and XHTML body of the request to be performed.

Assume you need to add signature for request client/statement. You will need to combine type of request (GET, POST, DELETE etc.) with URI of the request with all the parameters, unix timestamp and HTML request.

For example, to request account statement, combine request type GET with request URI and request parameters. Add unix timestamp at the end and include XML body if there is one for the request. Data for signature generation will be following:

```
GET/v1/client/statement?account=LV67RIBR00215170C0001&currency=EUR&fromDate=2025-01-02&toDate=2025-01-041746876398
```

To generate signature, you must complete following steps:

- Trim payload, removing non-printable characters \t, \r and \n.
- Create signature SHA512 with RSA.
- Encode the signature result to Base64.

Generated result shall be used in all requests in X-API-Signature header. For each request, new signature must be generated.

Example for account statement

Request type: GET

Request URI:

/v1/client/statement?account=LV67RIBR00215170C0001¤cy=EUR&fromDate=2025-01-02&toDate=2025-01-04

Timestamp: 1747382986

Payload for signature generation:

GET/v1/client/statement?account=LV67RIBR00215170C0001¤cy=EUR&fromDate=2025-01-02&toDate=2025-01-041747382986

Calculated signature:

aeEe/rfbs/NOqqIkxToVjEFCzz2ACNLIVQp2naM7vO9qy/U/mw/pMUT+0ZYkjair
FcAjZM2YYCt3i6/RP1DG80VENrNi8L1m7Tt7ncgMZIQ0fE2WkH+QMfm8ZB2K2p3
s8hYioA9v0Opszp2Pfw80klhg/yfjiiKyz19iSqZYOZINm7ktHfllVeZ29xslny6pxZR8
RVzGTWC7vbYS1CMAqgcZewNYDWIL1mT0tFyVduZJHYoOI/9VpzHtBpVsFGyXs
T4NXwe37jeFUZRi74NyCgvy3oQVUeZWxNZLR2j3Jq2q/1GuccUvPhcGxJ4TMZF
QjVqncO+Hsxr8diOtKXLhA==

Example payload for payment initiation

POST/v1/client/payment/import1747408362<?xml version=""1.0"" encoding=""UTF-8""?>

<Document

xmlns=""urn:iso:std:iso:20022:tech:xsd:pain.001.001.11"">

<CstmrCdtTrfInitn>

<GrpHdr>

<MsgId>MSG20241022145022845</MsgId>

<CreDtTm>2024-10-22T14:50:22</CreDtTm>

<NbOfTx>1</NbOfTx>

<CtrlSum>3001.38</CtrlSum>

<InitgPty>

<Nm>TEST USER</Nm>

<Id>

<OrgId>

<Othr>

<Id>1231231231</Id>

</Othr>

</OrgId>

</Id>

</InitgPty>

</GrpHdr>

<PmtInf>

<PmtInfId>186000</PmtInfId>

<PmtMtd>TRF</PmtMtd>

<BtchBookg>false</BtchBookg>

<NbOfTx>1</NbOfTx>

<CtrlSum>3001.38</CtrlSum>

<PmtTplnf>

<LclInstrm>

<Prtry>NORM</Prtry>

</LclInstrm>

</PmtTplnf>

<ReqdExctnDt>

```
<Dt>2025-05-16</Dt>
</ReqdExctnDt>
<Dbtr>
  <Nm>TEST USER</Nm>
  <PstlAdr>
    <Ctry>LV</Ctry>
    <AdrLine>Liepājas iela 2a, Rīga, LV-1029,
Latvija</AdrLine>
  </PstlAdr>
  <Id>
    <OrgId>
      <Othr>
        <Id>1231231231</Id>
      </Othr>
    </OrgId>
  </Id>
  <CtryOfRes>LV</CtryOfRes>
</Dbtr>
<DbtrAcct>
  <Id>
    <IBAN>LV40RIBR00215170N0000</IBAN>
  </Id>
</DbtrAcct>
<DbtrAgt>
  <FinInstnId>
    <BICFI>RIBRLV22</BICFI>
  </FinInstnId>
</DbtrAgt>
<ChrgBr>DEBT</ChrgBr>
<CdtTrfTxInf>
  <PmtId>
    <InstrId>24-19241</InstrId>
    <EndToEndId>24-19241</EndToEndId>
  </PmtId>
  <Amt>
    <InstdAmt Ccy=""EUR"">3001.38</InstdAmt>
  </Amt>
  <CdtrAgt>
    <FinInstnId>
      <BICFI>UNLALV2X</BICFI>
      <Nm>SEB banka AS</Nm>
      <PstlAdr>
        <AdrLine>SEB finanšu centrs, Meistaru
1, Valdlauči, Ķekavas novads, LV-1076</AdrLine>
      </PstlAdr>
    </FinInstnId>
  </CdtrAgt>
  <Cdtr>
    <Nm>TEST RECEIVER, SIA</Nm>
    <PstlAdr>
      <Ctry>LV</Ctry>
      <AdrLine>Peldu iela 10, Jelgava, LV-
3002</AdrLine>
    </PstlAdr>
  </Cdtr>
  <Id>
```

```

        <OrgId>
            <Othr>
                <Id>24234234</Id>
                <SchmeNm>
                    <Cd>TXID</Cd>
                </SchmeNm>
            </Othr>
        </OrgId>
    </Id>
    <CtryOfRes>LV</CtryOfRes>
</Cdtr>
<CdtrAcct>
    <Id>
        <IBAN>LV20UNLA0050004664251</IBAN>
    </Id>
</CdtrAcct>

    <RmtInf>
        <Ustrd>Payment for agreement No. 123123</Ustrd>
    </RmtInf>
</CdtTrfTxInf>
</PmtInf>
</CstmrCdtTrfInitn>
</Document>
```

Past days statement

Account information services for client's accounts in RIB supports requesting ISO20022 camt.052XML format statement.

This type of statement includes transactions for longer period in past (maximum 2 years from current date) up to current day. Data includes account balances for start date and end date of the statement. Date and entry sequence is oldest to newest.

Request method: GET

Request path:

```
/v1/client/statement?account={IBAN}&currency={CURRENCY}&fromDate={DATE_FROM}&toDate={DATE_TO}
```

Path parameter	Decription
IBAN	Account number for which statement is requested
CURRENCY	Currency for which statement is requested (EUR, USD)
DATE_FROM	Date from which statement is requested
DATE_TO	Date till which statement is requested

Request headers

Header	Decription
X-Api-Token	API token generated at user creation
X-Api-Signature	Calculated API signature
X-Timestamp	Timestamp used for API signature calculation

Example request

```
GET {URL}/v1/client/statement?
account=LV67RIBR00215170C0001&currency=EUR&fromDate=2025-01-
01&toDate=2025-10-15
```

Example response

```
<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:camt.052.001.02"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:iso:std:iso:20022:tech:xsd:camt.052.001.02camt.052.001.0
2.xsd">
```

```
<BkToCstmrAcctRpt>
  <GrpHdr>
    <MsgId>021517000220250510</MsgId>
    <CreDtTm>2025-05-10T14:27:07</CreDtTm>
  </GrpHdr>
  <Rpt>
    <Id>0215170002-EUR-0</Id>
    <CreDtTm>2025-05-10T14:27:07</CreDtTm>
    <FrToDt>
      <FrDtTm>2025-01-02T00:00:00</FrDtTm>
      <ToDtTm>2025-01-04T23:59:59</ToDtTm>
    </FrToDt>
    <Acct>
      <Id>
        <IBAN>LV67RIBR00123120C0001</IBAN>
      </Id>
      <Ccy>EUR</Ccy>
      <Ownr>
        <Nm>TEST USER</Nm>
        <Id>
          <PrvtId>
            <Othr>
              <Id>111111-11111</Id>
            </Othr>
          </PrvtId>
        </Id>
      </Ownr>
      <Svcr>
        <FinInstnId>
          <BIC>RIBRLV22</BIC>
        </FinInstnId>
      </Svcr>
    </Acct>
    <Bal>
      <Tp>
        <CdOrPrtry>
          <Cd>OPBD</Cd>
        </CdOrPrtry>
      </Tp>
      <Amt Ccy = "EUR">2095.15</Amt>
      <CdtDbtInd>CRDT</CdtDbtInd>
      <Dt>
        <Dt>2025-01-02</Dt>
      </Dt>
    </Bal>
    <Bal>
      <Tp>
        <CdOrPrtry>
          <Cd>CLBD</Cd>
        </CdOrPrtry>
      </Tp>
      <Amt Ccy = "EUR">1374.58</Amt>
      <CdtDbtInd>CRDT</CdtDbtInd>
      <Dt>
        <Dt>2025-01-04</Dt>
      </Dt>
      <Dt>2025-01-04</Dt>
    </Bal>
```

```
<TxsSummry>
  <TtlCdtNtries>
    <NbOfNtries>0</NbOfNtries>
    <Sum>0.00</Sum>
  </TtlCdtNtries>
  <TtlDbtNtries>
    <NbOfNtries>7</NbOfNtries>
    <Sum>720.57</Sum>
  </TtlDbtNtries>
</TxsSummry>
<Ntry>
  <NtryRef>202501021170154</NtryRef>
  <Amt Ccy="EUR">500.00</Amt>
  <CdtDbtInd>DBIT</CdtDbtInd>
  <Sts>BOOK</Sts>
  <BookgDt>
    <Dt>2025-01-02</Dt>
  </BookgDt>
  <ValDt>
    <Dt>2025-01-02</Dt>
  </ValDt>
  <AcctSvcrRef>202501021170154</AcctSvcrRef>
  <BkTxCd>
    <Prtry>
      <Cd>NTRF</Cd>
      <Issr>AS "TEST ENTITY"</Issr>
    </Prtry>
  </BkTxCd>
  <NtryDtls>
    <TxDtls>
      <Refs>
        <AcctSvcrRef>202501021170154</AcctSvcrRef>
        <InstrId>mk111</InstrId>
      </Refs>
      <AmtDtls>
        <TxAmt>
          <Amt Ccy="EUR">500.00</Amt>
        </TxAmt>
      </AmtDtls>
      <RltdPties>
        <Cdtr>
          <Nm>TEST USER</Nm>
          <PstlAdr>
            <Ctry>LV</Ctry>
          </PstlAdr>
          <Id>
            <OrgId>
              <Othr>
                <Id>111111-11111</Id>
              </Othr>
            </OrgId>
          </Id>
        </Cdtr>
        <CdtrAcct>
          <Id>
            <IBAN>LV36UNLA0050010020000</IBAN>
          </Id>
        </CdtrAcct>
      </RltdPties>
```

```
<RltdAgts>
  <CdtrAgts>
    <FinInstnId>
      <BIC>UNLALV2XXXX</BIC>
      <Nm>SEB BANKA</Nm>
    </FinInstnId>
  </CdtrAgts>
</RltdAgts>
<RmtInf>
  <Ustrd>Rekinu apmaksai un automatiskajiem maksajumiem -
sports, internets</Ustrd>
  </RmtInf>
</TxDtIs>
</NtryDtIs>
</Ntry>
<Ntry>
  <NtryRef>202501022780659</NtryRef>
  <Amt Ccy="EUR">35.09</Amt>
  <CdtDbtInd>DBIT</CdtDbtInd>
  <Sts>BOOK</Sts>
  <BookgDt>
    <Dt>2025-01-02</Dt>
  </BookgDt>
  <ValDt>
    <Dt>2025-01-02</Dt>
  </ValDt>
  <AcctSvcrRef>202501022780659</AcctSvcrRef>
  <BkTxCd>
    <Prtry>
      <Cd>NMSC</Cd>
      <Issr>AS "TEST ENTITY"</Issr>
    </Prtry>
  </BkTxCd>
  <NtryDtIs>
    <TxDtIs>
      <Refs>
        <AcctSvcrRef>202501022780659</AcctSvcrRef>
      </Refs>
      <AmtDtIs>
        <TxAmt>
          <Amt Ccy="EUR">35.09</Amt>
        </TxAmt>
      </AmtDtIs>
      <RltdPties>
        <Cdtr>
          <Nm>ABC Tehnika</Nm>
        </Cdtr>
      </RltdPties>
      <RmtInf>
        <Ustrd>Purchase (205) ABC Tehnika|*8129, TEST USER,30 Dec
2024,LVA 35.09 EUR</Ustrd>
      </RmtInf>
    </TxDtIs>
  </NtryDtIs>
</Ntry>
</Rpt>
</BkToCstmrAcctRpt>
</Document>
```

Payment initiation

API allows sending requests to RIB to fulfil payment orders from client's own current account and receive statuses for each individual payment.

Currently there are 3 endpoints for signed payment API:

- POST v1/client/payment/import: to import payment in XML pain.001.001.11 format
- GET v1/client/payment/status/{PAYMENT_ID}: to receive status of processing of the payment
- POST v1/client/payment/cancel: to cancel payment before processing has started.

Request method: POST

Request path:

v1/client/payment/import

Request headers

Header	Decription
X-API-Token	API token generated at user creation
X-API-Signature	Calculated API signature
X-Timestamp	Timestamp used for API signature calculation

Request body

Request body contains payment order in pain.001.001.11 format.

```
<?xml version="1.0" encoding="UTF-8"?>
<Document
  xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.11">
  <CstmrCdtTrfInitn>
    <GrpHdr>
      <MsgId>MSG20241022145022845</MsgId>
      <CreDtTm>2024-10-22T14:50:22</CreDtTm>
      <NbOfTx>1</NbOfTx>
      <CtrlSum>3001.38</CtrlSum>
      <InitgPty>
        <Nm>TEST USER</Nm>
        <Id>
          <OrgId>
            <Othr>
              <Id>1231231231</Id>
            </Othr>
          </OrgId>
        </Id>
      </InitgPty>
    </GrpHdr>

    <PmtInf>
```

Payment file import

```
<PmtInflId>186000</PmtInflId>
<PmtMtd>TRF</PmtMtd>
<BtchBookg>false</BtchBookg>
<NbOfTxs>1</NbOfTxs>
<CtrlSum>3001.38</CtrlSum>
<PmtTplnf>
  <LclInstrm>
    <Prtry>NORM</Prtry>
  </LclInstrm>
</PmtTplnf>
<ReqdExctnDt>
  <Dt>2025-05-16</Dt>
</ReqdExctnDt>
<Dbtr>
  <Nm>TEST USER</Nm>
  <PstlAdr>
    <Ctry>LV</Ctry>
    <AdrLine>Liepājas iela 2a, Rīga, LV-1029, Latvija</AdrLine>
  </PstlAdr>
  <Id>
    <OrgId>
      <Othr>
        <Id>1231231231</Id>
      </Othr>
    </OrgId>
  </Id>
  <CtryOfRes>LV</CtryOfRes>
</Dbtr>
<DbtrAcct>
  <Id>
    <IBAN>LV40RIBR00215170N0000</IBAN>
  </Id>
</DbtrAcct>
<DbtrAgt>
  <FinInstnId>
    <BICFI>RIBRLV22</BICFI>
  </FinInstnId>
</DbtrAgt>
<ChrgBr>DEBT</ChrgBr>
<CdtTrfTxInf>
  <PmtId>
    <InstrId>24-19241</InstrId>
    <EndToEndId>24-19241</EndToEndId>
  </PmtId>
  <Amt>
    <InstdAmt Ccy="EUR">3001.38</InstdAmt>
  </Amt>
  <CdtrAgt>
    <FinInstnId>
      <BICFI>UNLALV2X</BICFI>
      <Nm>SEB banka AS</Nm>
      <PstlAdr>
        <AdrLine>SEB finanšu centrs, Meistaru 1, Valdlauči, Ķekavas
novads, LV-1076</AdrLine>
      </PstlAdr>
    </FinInstnId>
  </CdtrAgt>
  <Cdtr>
```

Payment file import

```
<Cdtr>
  <Nm>TEST RECEIVER, SIA</Nm>
  <PstlAdr>
    <Ctry>LV</Ctry>
    <AdrLine>Peldu iela 10, Jelgava, LV-3002</AdrLine>
  </PstlAdr>
  <Id>
    <OrgId>
      <Othr>
        <Id>24234234</Id>
        <SchmeNm>
          <Cd>TXID</Cd>
        </SchmeNm>
      </Othr>
    </OrgId>
  </Id>
  <CtryOfRes>LV</CtryOfRes>
</Cdtr>
<CdtrAcct>
  <Id>
    <IBAN>LV20UNLA0050004664251</IBAN>
  </Id>
</CdtrAcct>

  <RmtInf>
    <Ustrd>Payment for agreement No. 123123</Ustrd>
  </RmtInf>
</CdtTrfTxInf>
</PmtInf>
</CstmrCdtTrfInitn>
</Document>
```

If payment was imported successfully, HTTP code 200 with empty response body is returned.

If payment initiation failed, error message is returned.

Example response

Example error response

```
{
  "status": "BAD_REQUEST",
  "message": "400 BAD_REQUEST \"PmtInf.ReqdExctnDt.Dt must not be in the past\"",
  "errors": [
    "400 BAD_REQUEST \"PmtInf.ReqdExctnDt.Dt must not be in the past\""
  ],
  "timestamp": "2025-05-16T17:52:19.407803549"
}
```

Payment status

This endpoint allows client to get processing status of the payment. In case the file processing fails in full, error code and message is returned and no payments are created. This can happen mainly due to invalid signature, lack of payment right or limits for account, XSD validation failure or error regarding pain.001 rules.

The return of status call is document in pain.002.002.12 (Customer Payment status report) format.

Request method: GET

Request path:

```
/v1/client/payment/status/{PAYMENT_ID}
```

Path parameter	Decription
PAYMENT_ID	Payment identification number from imported payment

Request headers

Header	Decription
X-Api-Token	API token generated at user creation
X-Api-Signature	Calculated API signature
X-Timestamp	Timestamp used for API signature calculation

Example request

```
GET {URL}/v1/client/payment/status/MSG20241022145022845
```

Response is returned in ISO20022 pain002.001.12 format.

Example response

```
<?xml version="1.0" encoding="UTF-8" ?>
<Doc:Document xmlns:Doc="urn:iso:std:iso:20022:tech:xsd:pain.002.001.12">
  <Doc:CstmrPmtStsRpt>
    <Doc:GrpHdr>
      <Doc:MsgId>15b9b6098f82400c954886decfb6382f</Doc:MsgId>
      <Doc:CreDtTm>2025-05-16T18:16:59.745+03:00</Doc:CreDtTm>
      <Doc:InitgPty>
        <Doc:Id>
          <Doc:OrgId>
            <Doc:AnyBIC>RIBRLV22</Doc:AnyBIC>
          </Doc:OrgId>
        </Doc:Id>
      </Doc:InitgPty>
```

Payment file import

```
</Doc:GrpHdr>
<Doc:OrgnlGrplnfAndSts>
  <Doc:OrgnlMsgId>MSG20241022145022845</Doc:OrgnlMsgId>
  <Doc:OrgnlMsgNmId>pain.001.001.11</Doc:OrgnlMsgNmId>
  <Doc:OrgnlNbOfTx>1</Doc:OrgnlNbOfTx>
  <Doc:OrgnlCtrlSum>3001.38</Doc:OrgnlCtrlSum>
</Doc:OrgnlGrplnfAndSts>
<Doc:OrgnlPmtlnfAndSts>
  <Doc:OrgnlPmtlnfId>186000</Doc:OrgnlPmtlnfId>
  <Doc:TxlnfAndSts>
    <Doc:OrgnlInstrId>24-19241</Doc:OrgnlInstrId>
    <Doc:OrgnlEndToEndId>24-19241</Doc:OrgnlEndToEndId>
    <Doc:TxSts>ACCP</Doc:TxSts>
  </Doc:TxlnfAndSts>
</Doc:OrgnlPmtlnfAndSts>
</Doc:CstmrPmtStsRpt>
</Doc:Document>
```

Example error response

```
{
  "status": "NOT_FOUND",
  "message": "404 NOT_FOUND \"Message SG20241022145022845 is not found\"",
  "errors": [
    "404 NOT_FOUND \"Message SG20241022145022845 is not found\""
  ],
  "timestamp": "2025-05-19T13:28:48.849903091"
}
```

Payment cancellation

This endpoint allows to cancel payment initiation requests for which processing hasn't started.

Request body shall contain payment cancellation message in ISO20022 camt.055.001.11 format.

Request method: POST

Request path:

```
/v1/client/payment/cancel
```

Request headers

Header	Decription
X-Api-Token	API token generated at user creation
X-Api-Signature	Calculated API signature
X-Timestamp	Timestamp used for API signature calculation

Payment file import

Example request body

```
<?xml version="1.0" encoding="utf-8"?>
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:camt.055.001.11" xmlns:xsi
="http://www.w3.org/2001/XMLSchema-instance">
  <CstmrPmtCxlReq>
    <Assgnmt>
      <Id>9874585</Id>
      <Assgnr>
        <Pty>
          <Nm>Maija Kemere</Nm>
        </Pty>
      </Assgnr>
      <Assgne>
        <Agt>
          <FinInstnId>
            <BICFI>RIBRLV22</BICFI>
          </FinInstnId>
        </Agt>
      </Assgne>
      <CreDtTm>2025-05-16T19:16:58.696</CreDtTm>
    </Assgnmt>
    <Undrlyg>
      <OrgnlGrplnfAndCxl>
        <OrgnlMsgId>MSG20241022145022845</OrgnlMsgId>
        <OrgnlMsgNmId>pain.001.001.11</OrgnlMsgNmId>
      </OrgnlGrplnfAndCxl>
    </Undrlyg>
  </CstmrPmtCxlReq>
</Document>
```

Example request

POST {URL}/v1/client/payment/cancel

Example request body

If payment was cancelled successfully, empty response is returned. Otherwise error is shown.

Example response

Example error response

```
{
  "status": "NOT_FOUND",
  "message": "404 NOT_FOUND \"Message MSG20241022145022845a is not found\"",
  "errors": [
    "404 NOT_FOUND \"Message MSG20241022145022845a is not found\""
  ],
  "timestamp": "2025-05-16T18:26:43.256105774"
}
```