

Bank Millennium

Millenet Link

Technical specification

12.09.2023

(changes are marked in yellow)



Table of contents

1. Introduction	2
1.1. Terminology and definitions.....	2
1.2. General description of Millenet Link Service	2
2. Launch and management of Millenet Link	3
2.1. Documentation	3
2.2. Communication certificates	3
2.3. Communication layer	3
2.3.1. An example of calling a service using .NET	3
2.4. Activities to be carried out in Millenet internet banking	8
2.4.1. Creation of the Millenet Link Virtual user	8
2.4.2. Supplementing the Virtual user data:	8
2.4.3. Loading the public key of the Customer's communication certificate	9
2.4.4. Activation form	10
2.4.5. Production / implementation test	10
2.4.6. Suspension / reactivation of Millenet Link communication	10
2.4.7. Preview / reference / download of the Customer's communication certificate	11
2.4.8. Prolongation of the Millenet Link Service	12
2.4.9. Dowloading Bank's certificate	12
2.4.10. Uploading the qualified certificate	12
3. Message formats	16
3.1. Bank accounts	17
3.1.1. GetAccount	17
3.1.2. GetAccountHistory	20
3.1.3. GetRecentAccountHistory	31
3.2. Statements	38
3.2.1. GetStatement	38
3.2.2. GetRecentStatement	41
3.3. Payments	44
3.3.1. CreditTransfer	44
3.3.2. GetPaymentStatusReport	85
3.4. Reports	88
3.4.1. GetReportList	88
3.4.2. GetReport	91
3.4.3. GetRecentReport	93
3.5. Factoring	96
3.5.1. InvoiceBatch	96
3.5.2. GetInvoiceBatchStatusReport	97
3.5.3. FactoringReports	99
3.6. Qualified signature	100
3.6.1. Functional description	100

1. Introduction

1.1. Terminology and definitions

- **Bank** - Bank Millennium S.A.
- **Bank's communication certificate** - a communication certificate provided by the Bank to the Customer, necessary to establish Client-Bank communication;
- **Customer (Client)** - The Bank's Corporate Customer with access to online banking
- **Customer's communication certificate** - a communication certificate issued by a trusted third party, provided by the Customer to the Bank, necessary to establish Client-Bank communication.
- **Millenet** - online banking, internet service for Bank's Customers
- **Millenet Link service** - solution that enables communication between the financial and accounting systems of the Bank's Clients and the Bank by means of network services (Web-services)
- **Virtual user** - technical user registered only within given company with no access to functionalities of Millenet. Browsing its details / modification of data is possible only from the level of the company's context.

1.2. General description of Millenet Link Service

The individual elements of the solution are based on the following standards:

- Recommendation of the Management Board of the Polish Bank Association regarding the adoption of a standard for the exchange of financial data between the Customer and the Bank and the Bank and the Customer on the Polish market from November 2010 - hereinafter referred to as RBE,
- Common platform for the financial industry for the development of standard XML syntax messages developed by the International Organization for Standardization <http://www.iso20022.org/> - hereinafter referred to as ISO 20022.

The Millenet Link service offers the Customers the possibility of a much faster and more efficient exchange of information between ERP/FA systems operating within the company and the electronic banking system. Thanks to the service, it is possible to download current financial data for ERP/FA systems from the level of these systems, without the need to import/export files containing this data between the electronic banking system and accounting systems. This solution significantly speeds up the exchange of information between the Bank and the Customer, as well as simplifies the information exchange process. The mechanism of sending orders by the Customer directly from the ERP/FA systems is simplified in the same way.

The exchange of data between the Bank and the Customer in the Millenet Link service takes place through the exchange of messages in the XML file format using the web-services technology. All communication is secured by a secure HTTPS channel using double-sided SSL.

2. Launch and management of Millenet Link

2.1. Documentation

In order to start the Millenet Link service it is necessary:

- submit an application to start the Millenet Link service
- read and accept the Millenet Link Regulations
- meet the requirements described in this Technical Specification
- activate the service, which involves signing and submitting the Millenet Link Activation Form to the Bank.

2.2. Communication certificates

The Millenet Link service is based on communication secured by communication certificates. Bank accepts certificates issued by companies recommended by the National Bank of Poland, detailed information are available on the website <https://www.nccert.pl/>.

There are two types of communication certificates in the service structure:

- Customer's communication certificate,
- Bank's communication certificate,

The private key of the Customer's communication Certificate is installed on the Client's side in an environment designed to exchange communication with the Bank.

The public key of the Customer's communication Certificate is transferred to the Bank via the Millenet system, which is more in point 2.4.3.

Minimum requirements of the certificate parameters: X.509 standard, unqualified, PEM format, RSA key length minimum 2048 bits, signed by the Trusted Third Party, with application for Client authentication.

2.3. Communication layer

All data transmitted between the Bank's Customer and the Bank's infrastructure use the HTTP protocols 1.1 and the SOAP protocol, secured on both sides by authenticated TLS (Transport Layer Security) protocol. Validity of the certificate will be verified using the OCSP (Online Certificate Status Protocol). The solution assumes no encryption at the message level.

Communication takes place in the Client-server model, where the Bank's system serves as a server, which means that communication is always initiated by the Client. Client sends a SOAP request with a message compliant with the POST method described in this document.

Each call to the Bank's service begins with the initialization of the double-sided TLS tunnel.

The customer can call from a maximum of 4 different IP addresses or, alternatively, a maximum of 4 different IP address classes assigned to users of the Millenet Link service. IP addresses should be defined in Millenet.

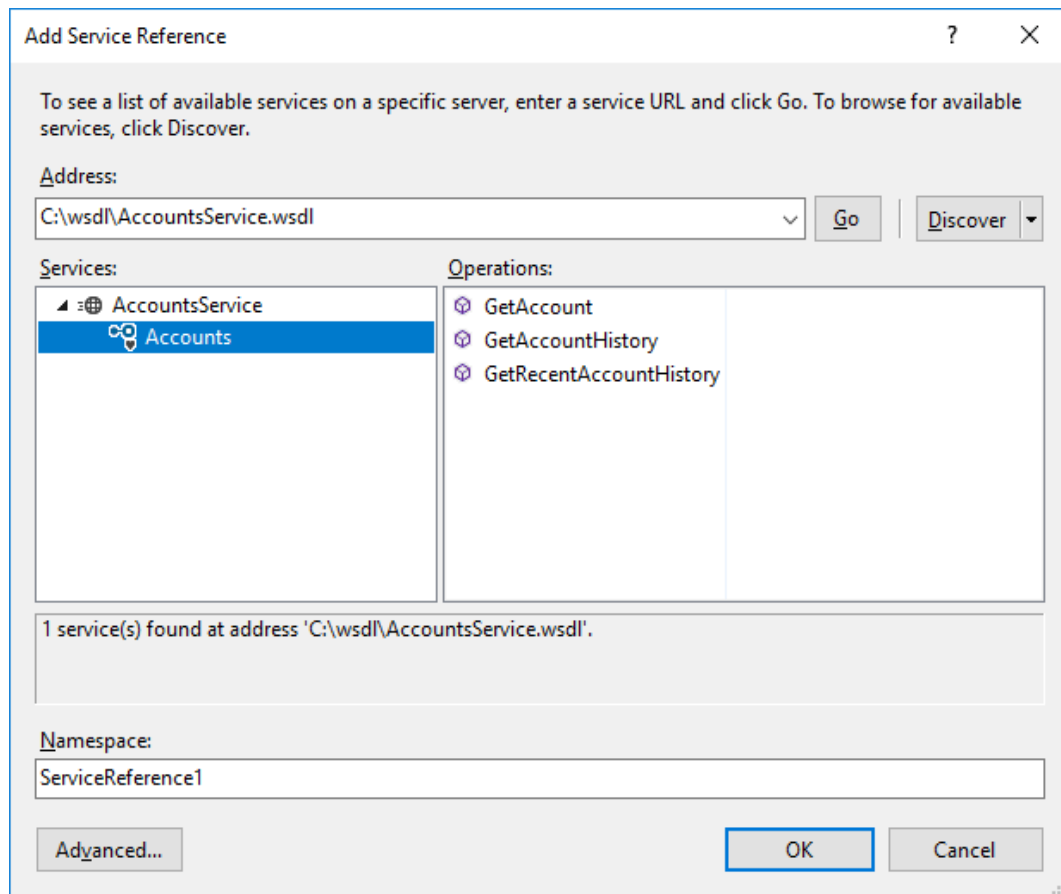
2.3.1. An example of calling a service using .NET

This chapter describes the step-by-step rules for creating a Client for Millenet Link services on the .NET platform using the Visual Studio 2015 development environment.

The services in the Millenet Link service are grouped into business categories. Available methods are described in p. 3.

The process of generating the client for each of the services is analogous, so the example will be based on one of the services: *GetAccount*.

Creating the Client is best to be done with the help of the Visual Studio application wizard. To do this, use the Project -> Add Service Reference menu in an open project in Visual Studio. In the dialog box we give the location of the WSDL file. WSDL files should be provided with the instructions.



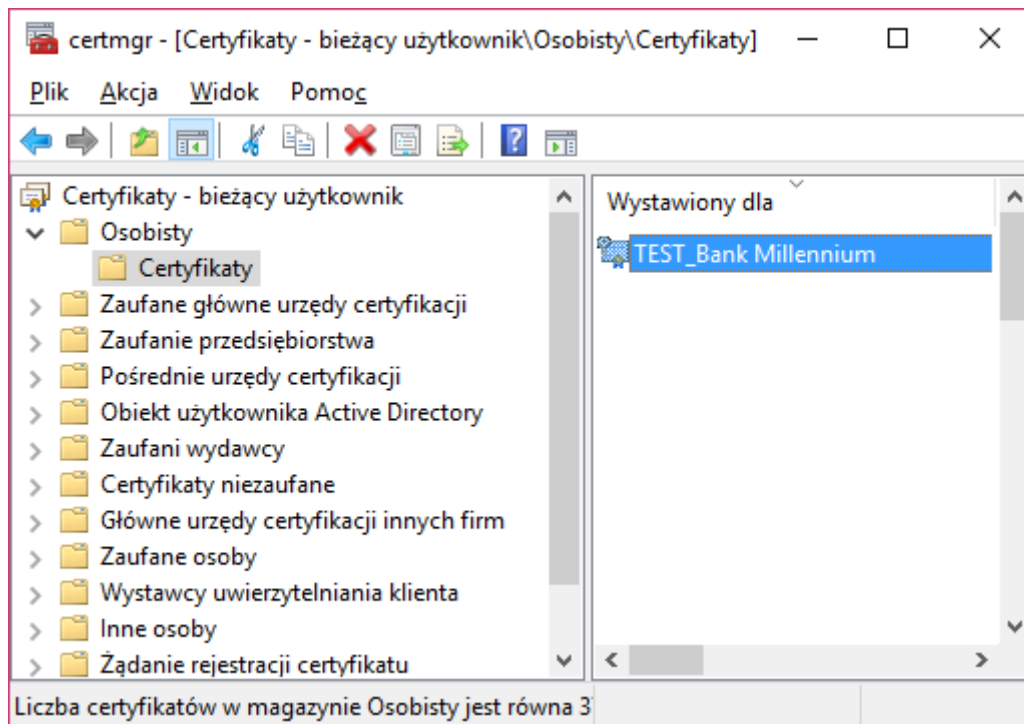
After clicking OK the classes corresponding to the service will be generated, in the example above, related to the accounts.

The next step in the connection configuration is to modify the app.config application configuration file generated by the wizard by default.

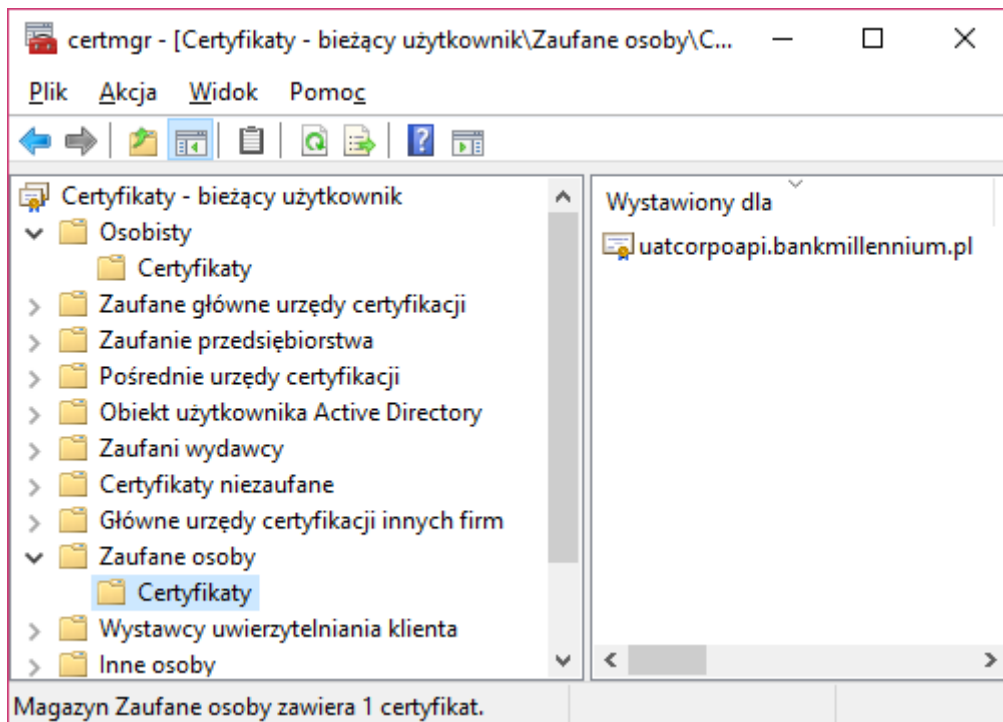
Configuration in app.config file

Configuration generated by default in the app.config file does not contain all the elements needed to set up the connection, such as behaviorConfiguration, describing the rules for establishing an encrypted connection. The customBinding element is also missing.

The Client's certificate (clientCertificate) with private and public keys should have been previously imported into the Windows user certificate store.



The server's certificate (serviceCertificate), which can be downloaded by Millenet, should be imported into the Trusted People folder:



The first step in configuring the app.config will be to define the <behavior> element. In this section, define the Client's cryptographic key (clientCertificate) and server key (serviceCertificate), both installed on the machine from which the connection will be made.

```
<behaviors>
  <endpointBehaviors>
    <behavior name="sslBehavior">
      <clientCredentials>
```

```

        <clientCertificate findValue="[NUMER_SERYJNY_KLUCZA_KLIENTA]"
storeLocation="CurrentUser" storeName="My" x509FindType="FindBySerialNumber"/>
        <serviceCertificate>
            <defaultCertificate findValue="[ODCISK_PALCA_KLUCZA_SERWERA]"
storeLocation="CurrentUser" storeName="TrustedPeople" x509FindType="FindByThumbprint"/>
        </serviceCertificate>
    </clientCredentials>
</behavior>
</endpointBehaviors>
</behaviors>

```

The second important element is the <binding> section, to which <customBinding> should be added. You can delete the automatically generated basicHttpBinding:

```

<bindings>
    <basicHttpBinding>
        <binding name="BasicHttpBinding_Statements" />
    </basicHttpBinding>
    <customBinding>
        <binding name="WsHttpBinding">
            <textMessageEncoding messageVersion="Soap11" />
            <httpsTransport requireClientCertificate="true" />
        </binding>
    </customBinding>
</bindings>

```

The last element of the configuration in the app.config file is the modification of the <client> section so that individual endpoints use the previously defined *customBinding* and *behaviors*.

```

<endpoint address=" https://millenetlink.bankmillennium.pl/AccountsService.svc "
        behaviorConfiguration="sslBehavior" binding="customBinding"
        bindingConfiguration="WsHttpBinding" contract="Accounts.Accounts"
        name="BasicHttpBinding_Accounts" />

```

In case the configuration is refreshed by using the UpdateServiceReference option, the above changes in the app.config file should be performed again, because the Visual Studio setup wizard will override them with default values.

An example of Client the GetAccount service

Below is the complete source code for the GetAccount service:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CorpoApiClient
{
    using Accounts;

```

```

class Program
{
    static void Main(string[] args)
    {
        using (var client = new AccountsClient())
        {
            var response = client.GetAccount(new Document2()
            {
                GetAcct = new GetAccountV04()
                {
                    MsgId = new MessageIdentification()
                    {
                        Id = Guid.NewGuid().ToString()
                    },
                    AcctQryDef = new AccountQueryDefinition4()
                    {
                        AcctCrit = new AccountCriteriaDefinition4Choice()
                        {
                            Item = new AccountCriteria4()
                            {
                                SchCrit = new[]
                                {
                                    new CashAccountSearchCriteria4()
                                    {
                                        AcctId =
                                            new[]
                                            {
                                                new AccountIdentificationSearchCriteriaChoice()
                                                {
                                                    ItemElementName = ItemChoiceType15.EQ,
                                                    Item = new AccountIdentification1Choice() {
ItemElementName = ItemChoiceType14.IBAN, Item = "PL92116022020000000111379980" }
                                                },
                                            },
                                        Bal = new[] { new BalanceDetails4()
                                        {
                                            BalTp = new[] { new BalanceType3Choice() { Item =
BalanceType10Code.AVLB }, }, CtrPtyTp = BalanceCounterparty1Code.MULT
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

});

Console.WriteLine(response);
}
}
}
}
}

```

2.4. Activities to be carried out in Millenet internet banking

After submitting the Application for starting the service, the Bank will provide information about the possibility of starting the Millenet Link service configuration in the Millenet system.

2.4.1. Creation of the Millenet Link Virtual user

Go to the Millenet Link management module. Authorized users after logging in to Millenet should go to Settings -> Millenet Link Management -> New Virtual user.

2.4.2. Supplementing the Virtual user data:

Login	Example login
Description	Example description
Adres IP	The address from which the connection to the Bank will be made You can define up to 4 different IP addresses or, alternatively, up to 4 different classes of IP addresses from which a connection to the Bank will be made. The structure of the IP class is X.X.X.0, where X is an 8-bit number.

Permissions and accounts

Permission	Method
Account information	GetAccount
	GetAccountHistory
	GetRecentAccountHistory
Payments	CreditTransfer
	GetPaymentStatusReport
Statements	GetStatement
	GetRecentStatement
Reports	GetReportList
	GetReport
	GetRecentReport
Factoring	FactoringReports
	GetInvoiceBatchStatusReport
	InvoiceBatch

It should be defined to which accounts the Virtual user will have access and at what level:

- View - without the possibility of making transactions - the function such as CreditTransfer will not work
- Edit - with the possibility of making transactions

Add new user

Login

Description

IP address - +

IP address class (x.x.): - +

IP address - +

IP address class (x.x.): - +

Methods

☒ Account information

☒ Payments

☒ Reports

☒ Statements

Accounts

Account number	☒ View	☒ Edit
96116022020000000113616631	☒	☒
58116022020000000113716740	☒	☒
51116022020000000113716769	☒	☒

The last step in the process of creating a Virtual user is selecting the Save button. The entire operation does not require additional approval.

The user is visible on the Virtual users List with the status "Pending for activation".

2.4.3. Loading the public key of the Customer's communication certificate

Loading the public key of the Client's communication certificate should be performed in Settings -> Millenet Link Management -> Details on a given User -> Certificates -> Upload.

Virtual user details

Details Certificates Permissions

Serial Number Status Alias

Select date

From dd / mm / yyyy

To dd / mm / yyyy

Alias	Serial Number	Expiration Date	Status	Actions
1				

Back to list Upload Download Bank's certificate

Please upload the key with the *.cer extension and assign Alias (any description).

2.4.4. Activation form

After uploading the certificate in the Virtual users List, the print form icon will appear next to the given user. The form will be automatically completed by the system in accordance with the entered data and the permissions granted.

The document should be printed, signed in accordance with the representation (or separate proxies) and then delivered to the Bank.

After the document has been processed, the Bank activates the Millenet Link service.

If the status of the Virtual user from "Pending for activation" changes to "Active" status, the Millenet Link connection will be active and ready for testing.

2.4.5. Production / implementation test

Scenario and methods of implementation tests implementation are determined individually between the Bank and the Customer.

2.4.6. Suspension / reactivation of Millenet Link communication

In order to suspend communication possibilities, please go to Settings -> Millenet Link Management -> Details on a given User -> Suspend communication.

The suspension is temporary and can be revoked. To reactivate communication, go to Settings -> Millenet Link Management -> Details on a given user -> Reactivate communication.

Virtual user details

Details

Certificates

Permissions

Login

Description

IP address

IP address

IP Class

Status Suspended

Reactivate access

Back to list

Edit

2.4.7. Preview / reference / download of the Customer's communication certificate

Certificate preview: Settings -> Millenet Link Management -> Details on a given User -> Certificates -> Actions, Details on a given certificate.

Virtual user details

Details

Certificates

Permissions

Serial Number

Status

Alias

Select date

From dd / mm / yyyy

To dd / mm / yyyy

Alias	Serial Number	Expiration Date	Status	Actions
		0 / / 14	Active	Details Activation form

1

10

Back to list

Upload

Download Bank's certificate

The revocation of the Customer's communication certificate is an irreversible process. If the certificate is revoked, a new certificate should be purchased and a new Virtual User should be set up. If you need to revoke the certificate, go to Settings -> Millenet Link Management -> Details on a given User -> Certificates -> Details on a given certificate -> Revoke certificate.

Certificate details

Alias [REDACTED]

Full name

Expiration date 2020-03-05T13:00:04

Issuer CN=Certum Digital Identification CA SHA2, OU=Certum Certification Authority, O=Unizeto Technologies S.A., C=PL

Serial Number 13 [REDACTED] A

Status 1

Subject E=wydruki@bankmillennium.pl, CN=mille01-TEST, L=Warszawa, OU=DBC, O=Bank Millennium SA, C=PL

Thumbprint 4482 [REDACTED] b4

[Back to list](#) [Revoke certificate](#) [Download certificate](#)

Downloading the certificate: Settings -> Millenet Link Management -> Details on a given User -> Certificates -> Details on a given certificate -> Download certificate.

2.4.8. Prolongation of the Millenet Link Service

Each Customer's communication certificate has an expiration date. At least 14 days before the certificate expires, a new Virtual user should be created and all steps described in point 2.4.1-2.4.4. should be executed.

2.4.9. Downloading Bank's certificate

The Customer may at any time download the Bank's certificate in order to authenticate the connection.

To do this, follow these steps: Settings -> Millenet Link Management -> Details on a given virtual user -> Certificates -> Download Bank's certificate

Virtual user details

[Details](#) [Certificates](#) [Permissions](#)

Serial Number Status Alias

Select date

From dd / mm / yyyy

To dd / mm / yyyy

Alias	Serial Number	Expiration Date	Status	Actions
1				

[Back to list](#) [Upload](#) [Download Bank's certificate](#)

2.4.10. Uploading the qualified certificate

Using the Millenet Link service, Customer can also make transfers directly in internal system, after signing them with a qualified certificate. To use this option, please follow these steps:

- Use the qualified certificates tab after entering Menu-> Settings-> Millenet Link Management and the "Import certificate" option

Virtual users **Qualified certificates**

Details of authorization tools for Millenet Link

Status: All Login: Certificate name: Expiration Date: dd / mm / yyyy

Login	Name and surname	Accept group	Certificate name	Status	Expiration Date	Operations
1						

10

Import certificate

- b) A list of users with authorization to authorize operations will be expanded. Select the user for whom the public key of the qualified certificate will be imported into the system, and then specify the location of the key and import it.

Users with possibility of importing qualified certificate

Login:

Login	Name and surname	Accept group	
		A	+ Import certificate
		A	+ Import certificate
		A	- Fold

File

Alias:

Upload certificate

		A	+ Import certificate
		A	+ Import certificate
		A	+ Import certificate
		C	+ Import certificate
		A	+ Import certificate
		A	+ Import certificate
		G	+ Import certificate

1 2 3 4 5 6 Next >

10

- c) After importing, the certificate is active and you can perform the following actions on it:
1. DETAILS - the option is available only for the owner of the certificate - contains the details of the certificate, its owner and enables the following actions:
 - i. Suspend / Resume certificate - action suspending the operation of the certificate, possible to withdraw (renew the certificate) without the Bank's interference
 - ii. Lock certificate - unlocking the certificate can only be done by a Bank employee
 - iii. Delete certificate - after deleting it will not be possible to upload the same certificate again

- iv. Download certificate - download the previously qualified public key of a qualified certificate
2. SUSPEND / RESUME - action suspending certificate operation, possible to withdraw (certificate renewal) without Bank intervention

Virtual users
Qualified certificates

Details of authorization tools for Millenet Link

Status
All
Login
Certificate name
Expiration Date
dd / mm / yyyy

Login	Name and surname	Accept group	Certificate name	Status	Expiration Date	Operations
		A		Active	14/05/2021 12:00:00	Deactivate Details

1
10

Import certificate

Certificate details

Alias test
Full name
Expiration date 14/05/2021 12:00:00
Issuer OID.2.5.4.97=VATPL CN=COPE SZAFIR - Kwalifikowany, O=Krajowa Izba Rozliczeniowa S.A., C=PL
Serial Number 45
Status Active
Subject SN= G= CN= SERIALNUMBER=PNOPL- C=PL
Thumbprint c7f0

Back to list
Revoke
Block
Remove
Download

History

Action executed by:	Date	Previous status	New status
	30/05/2021 12:52	Inactive	Active
	30/05/2021 14:48	Active	Inactive

Show more

- d) Next, modify the permissions for the Virtual User in the Menu-> Settings-> Millenet Link Management-> Virtual Users-> Details of the selected Virtual User-> Permissions and allow authorization with a qualified certificate transaction

Virtual user details

Details

Certificates

Permissions

Methods

☒ Account information

☒ Payments

☒ Reports

☒ Statements

☒ Factoring

Authorization section

☒ Allow authorization with qualified certificate

Accounts

Account number	View	Edit
----------------	------	------

e) The changes required in the Customer's internal system are described in Chapter 3

3. Message formats

As part of the Millenet Link service, the following list of messages, described in details in the subsequent chapters of the manual, is available.

The number of the instruction manual	The name of the message
3.1. Bank accounts	Address: https://millenetlink.bankmillennium.pl/AccountsService.svc
3.1.1.1	GetAccountRequest
3.1.1.2	GetAccountResponse
3.1.2.1	GetAccountHistoryRequest
3.1.2.2	GetAccountHistoryResponse
3.1.3.1	GetRecentAccountHistoryRequest
3.1.3.2	GetRecentAccountHistoryResponse
3.2. Statements	Address: https://millenetlink.bankmillennium.pl/StatementsService.svc
3.2.1.1	GetStatementRequest
3.2.1.2	GetStatementResponse
3.2.2.1	GetRecentStatementRequest
3.2.2.2	GetRecentStatementResponse
3.3. Payments	Address: https://millenetlink.bankmillennium.pl/PaymentsService.svc
3.3.1	CreditTransfer
3.3.1.1.1	Domestic
3.3.1.1.2	Express
3.3.1.1.3	Sorbnnet
3.3.1.1.4	Internal
3.3.1.1.5	Foreign
3.3.1.1.6	TAX
3.3.1.1.7	Social
3.3.1.1.8	Postal order
3.3.1.1.9	Prepaid - transfer from prepaid card account
3.3.1.2	CreditTransferResponse
3.3.2.1	GetPaymentStatusReportRequest
3.3.2.2	GetPaymentStatusReportResponse
3.4. Reports	Adress: https://millenetlink.bankmillennium.pl/ReportsService.svc
3.4.1.1	GetReportListRequest
3.4.1.2	GetReportListResponse
3.4.2.1	GetReportRequest
3.4.2.2	GetReportResponse
3.4.3.1	GetRecentReportRequest
3.4.3.2	GetRecentReportREsponse
3.5. Factoring	Adress: https://millenetlink.bankmillennium.pl/FactoringService.svc
3.5.1.1	InvoiceBatchRequest
3.5.1.2	InvoiceBatchResponse

3.5.2.1	GetInvoiceBatchStatusReport
3.5.2.2	GetInvoiceBatchStatusResponse

3.1. Bank accounts

3.1.1. GetAccount

The GetAccountHistory method supports the exchange of information between the Customer and the Bank regarding the amount of the balance on the Client's account.

3.1.1.1. GetAccountRequest

Sent by the Customer to the Bank. It contains parameters allowing to identify the account and balance parameters that the Customer wants to ask.

Message structure:

Field name	XML element name	Status	Format	Comments
Message ID	GetAcct +MsgId ++Id	O	String	Unique for a given recipient. Can not contain spaces.
Account number	GetAcct +AcctRptQryDef ++AcctRptCrit +++NewCrit ++++SchCrit +++++AcctId +++++EQ +++++IBAN +++++BBAN	M	String(30)	Acceptable types of identifiers the account: IBAN, BBAN. The maximum field size is 30n.
Balance type	GetAcct +AcctQryDef ++AcctCrit +++NewCrit ++++SchCrit +++++Bal +++++BalTp +++++Cd	M	String(4)	Valid only values: 'AVLB' - available funds 'BOOK' - accounting balance The element can occur twice if we want to download both types of balances at once.
Multilateral balance	GetAcct +AcctQryDef ++AcctCrit +++NewCrit ++++SchCrit +++++Bal +++++CtrPtyTp	M	String(4)	The only valid value is 'MULT'

Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://www.bankmillennium.pl/CorporateServices" xmlns:urn="urn:swift:xsd:camt.003.001.04">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:GetAccount>
      <tem:request>
        <urn:GetAcct>
          <urn:MsgId>
            <urn:Id>Identyfikator wiadomości</urn:Id>
          </urn:MsgId>
          <urn:AcctQryDef>
            <urn:AcctCrit>
              <urn:NewCrit>
                <urn:SchCrit>
                  <urn:AcctId>
                    <urn:EQ>
                      <urn:IBAN>Bank account number</urn:IBAN>
                    </urn:EQ>
                  </urn:AcctId>
                  <urn:Bal>
                    <urn:BalTp>
                      <urn:Cd>AVLB</urn:Cd>
                    </urn:BalTp>
                    <urn:CtrPtyTp>MULT</urn:CtrPtyTp>
                  </urn:Bal>
                  <urn:Bal>
                    <urn:BalTp>
                      <urn:Cd>BOOK</urn:Cd>
                    </urn:BalTp>
                    <urn:CtrPtyTp>MULT</urn:CtrPtyTp>
                  </urn:Bal>
                </urn:SchCrit>
              </urn:NewCrit>
            </urn:AcctCrit>
          </urn:AcctQryDef>
        </urn:GetAcct>
      </tem:request>
    </tem:GetAccount>
  </soapenv:Body>
</soapenv:Envelope>
```

3.1.1.2. GetAccountResponse

Sent by the Bank to the Customer. It contains information about the balance on the Customer's account indicated in the inquiry.

Message structure:

Field name	XML element name	Status	Format	Comments
File order number	RtrAcct +MsgId ++Id	O	String	Unique for a given recipient. Can not contain spaces
Account number	RtrAcct +BizRpt ++AcctRpt +++AcctId	M	String(30)	Acceptable types: IBAN, BBAN
Account currency	RtrAcct +BizRpt ++AcctRpt +++Acct ++++Ccy	M	String(3)	3!a
Multilateral balance	RtrAcct +BizRpt ++AcctRpt +++Acct ++++MulBal	M	-	The element can occur twice
Balance amount	RtrAcct +BizRpt ++AcctRpt +++Acct ++++MulBal +++++Amt	M	Number(*.2)	
Balance indicator	RtrAcct +BizRpt ++AcctRpt +++Acct ++++MulBal +++++CdtDbtInd	M	String(4)	Acceptable values 'CRDT' - for balance ≥ 0 'DBIT' - for balance < 0
Balance type	RtrAcct +BizRpt ++AcctRpt +++Acct ++++MulBal +++++Tp +++++Cd	M	String(4)	Valid only values: 'AVLB' - available funds 'BOOK' - accounting balance The element can occur twice if we want to get both types of balances at once.

Example:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <GetAccountResponse xmlns="http://www.bankmillennium.pl/CorporateServices">

```

```

<GetAccountResult>
  <RtrAcct xmlns="urn:swift:xsd:camt.004.001.04">
    <MsgId>
      <Id>Message ID</Id>
    </MsgId>
    <BizRpt>
      <AcctRpt>
        <AcctId>
          <IBAN> Bank account number </IBAN>
        </AcctId>
        <Acct>
          <Ccy>Currency</Ccy>
          <MulBal>
            <Amt>Amount</Amt>
            <CdtDbtInd>CRDT</CdtDbtInd>
            <Tp>
              <Cd>AVLB</Cd>
            </Tp>
          </MulBal>
        </Acct>
      </AcctRpt>
      <AcctRpt>
        <AcctId>
          <IBAN> Bank account number </IBAN>
        </AcctId>
        <Acct>
          <Ccy>Currency</Ccy>
          <MulBal>
            <Amt>Amount</Amt>
            <CdtDbtInd>CRDT</CdtDbtInd>
            <Tp>
              <Cd>BOOK</Cd>
            </Tp>
          </MulBal>
        </Acct>
      </AcctRpt>
    </BizRpt>
  </RtrAcct>
</GetAccountResult>
</GetAccountResponse>
</s:Body>
</s:Envelope>

```

3.1.2. GetAccountHistory

The GetAccountHistory method supports the exchange of information between the Customer and the Bank regarding the history of operations on the Client's account.

3.1.2.1. GetAccountHistoryRequest

Sent by the Customer to the Bank. It contains parameters that allow identifying the account and the time for which the Client expects history.

Struktura wiadomości:

Field name	XML element name	Status	Format	Comments
Message ID	GetAcctRpt +MsgId ++Id	O	String(30)	Unique for a given recipient. Can not contain spaces
History pagination	GetAcctRpt +GrpHdr ++Msgpgntn +++PgNb	C	String	If the history search period has more than 600 records, the account history page number is received in the GetAccountHistory response, always entered in the second query in the given download queue
Account number	GetAcctRpt +AcctRptQryDef ++AcctRptCrit +++NewCrit ++++SchCrit +++++AcctId ++++++EQ +++++++IBAN	M	String(30)	Account number in IBAN format
The starting date for history	GetAcctRpt +AcctRptQryDef ++AcctRptCrit +++NewCrit ++++SchCrit +++++AcctRptValDt ++++++DtSch +++++++FrDt	M	ISO Date	
End date for history	GetAcctRpt +AcctRptQryDef ++AcctRptCrit +++NewCrit ++++SchCrit +++++AcctRptValDt ++++++DtSch +++++++ToDt	M	ISO Date	

Example:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://www.bankmillennium.pl/CorporateServices"
xmlns:urn="http://www.bankmillennium.pl/CorporateServices">
<soapenv:Header/>
<soapenv:Body>
  <tem:GetAccountHistory>
    <tem:request>
      <urn:GetAcctRpt>
        <urn:MsgId>
          <urn:Id>Message ID</urn:Id>
        </urn:MsgId>
        <urn:AcctRptQryDef>
          <urn:AcctRptCrit>
            <urn:NewCrit>
              <urn:SchCrit>
                <urn:AcctId>
                  <urn:EQ>
                    <urn:IBAN> Bank account number </urn:IBAN>
                  </urn:EQ>
                </urn:AcctId>
                <urn:AcctRptValDt>
                  <urn:DtSch>
                    <urn:FrDt>Date from</urn:FrDt>
                    <urn:ToDt>Date to</urn:ToDt>
                  </urn:DtSch>
                </urn:AcctRptValDt>
              </urn:SchCrit>
            </urn:NewCrit>
          </urn:AcctRptCrit>
        </urn:AcctRptQryDef>
      </urn:GetAcctRpt>
    </tem:request>
  </tem:GetAccountHistory>
</soapenv:Body>
</soapenv:Envelope>

```

Example with pagination:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://www.bankmillennium.pl/CorporateServices"
xmlns:urn="http://www.bankmillennium.pl/CorporateServices">
<soapenv:Header/>
<soapenv:Body>
  <tem:GetAccountHistory>
    <tem:request>
      <urn:GetAcctRpt>

```

```

<urn:MsgId>
  <urn:Id>Message ID</urn:Id>
</urn:MsgId>
<urn:GrpHdr>
  <urn:MsgPgntn>
    <urn:PgNb>-History pagination - number</urn:PgNb>
    </urn:MsgPgntn>
  </urn:GrpHdr>
<urn:AcctRptQryDef>
  <urn:AcctRptCrit>
    <urn:NewCrit>
      <urn:SchCrit>
        <urn:AcctId>
          <urn:EQ>
            <urn:IBAN> Bank account number </urn:IBAN>
          </urn:EQ>
        </urn:AcctId>
      <urn:AcctRptValDt>
        <urn:DtSch>
          <urn:FrDt>Date from</urn:FrDt>
          <urn:ToDt>Date to</urn:ToDt>
        </urn:DtSch>
      </urn:AcctRptValDt>
    </urn:SchCrit>
  </urn:NewCrit>
</urn:AcctRptCrit>
</urn:AcctRptQryDef>
</urn:GetAcctRpt>
</tem:request>
</tem:GetAccountHistory>
</soapenv:Body>
</soapenv:Envelope>

```

3.1.2.2. GetAccountHistoryResponse

Sent by the Bank to the Customer. It contains the details of operations that took place in a given account in the time indicated in the query.

Message structure:

Field name	XML element name	Status	Format	Comments
Message ID	BkToCstmrAcctRpt +GrpHdr ++MsgId	O	String(30)	Unique for a given recipient. Can not contain spaces
Date and time of creation	BkToCstmrAcctRpt +GrpHdr ++CreDtTm	M	ISO DateTime	ISO DateTime

History pagination	BkToCstmrAcctRpt +GrpHdr ++MsgPgntn	C	String	The unique history pagination number, made available when 600 records are exceeded in the requested time period, the number for pasting to subsequent inquiries for the requested account history
History pagination - the last page	BkToCstmrAcctRpt +GrpHdr ++MsgPgntn	M	String	It shows whether this is the last answer to the desired account history. Possible answers are: 'False' - means that after the query the next pagination number will be sent 'True' - last on the history page
Message ID	BkToCstmrAcctRpt +Rpt ++Id	M	String(30)	The same ID as in the header
Date and time of creation	BkToCstmrAcctRpt +Rpt ++CreDtTm	M	ISO DateAndTime	The same date as in the header
Starting date and time for history	BkToCstmrAcctRpt +Rpt ++FrToDt +++FrDtTm	M	ISO DateAndTime	Only the date is significant, the time is always set to 00:00
End date and time for history	BkToCstmrAcctRpt +Rpt ++FrToDt +++ToDtTm	M	ISO DateAndTime	Only the date is significant, the time is always set to 00:00
Account number	BkToCstmrAcctRpt +Rpt ++Acct +++Id ++++IBAN	M	String(30)	Account number in IBAN format
Item in history	BkToCstmrAcctRpt +Rpt ++Ntry	O	-	It can occur 0 or many times
Fields in the history item (inside the <Ntry> tag):				
Item amount	+Amt	M	String(3)+Number(*,2)	The amount and currency of the operation. Mandatory entering also the currency code (ISO).

Item indicator	+CdtDbtInd	M	String(4)	Acceptable values 'CRDT' - for the amount ≥ 0 'DBIT' - for the amount <0
Item status	+Sts	M	String(4)	Acceptable value BOOK.
Accounting date	+BookgDt ++DtTm	M	ISO DateAndTime	If there is no information about the accounting time, it will be returned in the form of 00:00
Operation date	+ValDt ++DtTm	M	ISO DateAndTime	If there is no information about the operation time, it will be returned in the form of 00:00
Banking operation code	+BkTxCd ++Domn +++Cd	M	String	Standard operating code returned by the banking system (list of basic codes is included below)
Extended operation code	+BkTxCd ++Domn +++Fmly ++++Cd	O	String	
Extended operation subcode	+BkTxCd ++Domn +++Fmly ++++SubFmlyCd	O	String	
Transaction number during the day	+NtryDtls ++TxDtls +++Refs ++++Msgld	O	String	Number for downloading incremental history
Contractor	+NtryDtls ++TxDtls +++RltdPties ++++Cdtr	O	-	For debit operations (details of the marker included in the next table)
Name of the contractor	+NtryDtls ++TxDtls +++RltdPties ++++Cdtr ++++Nm	O	String	Contractor's details (Name, Address, city, zip code)
Contractor's account	+NtryDtls ++TxDtls +++RltdPties ++++CdtrAcct ++++Id ++++Othr	O	String(30)	For debit operations The account number (regardless of whether IBAN or not) is returned in the element

	+++++Id			Account is not shown in Balancing VAT transactions
Transfer title - unstructured	+NtryDtls ++TxDtls +++RmtInf ++++Ustrd	O	String(140)	Operation title (unstructured)
Transfer title - structured	+NtryDtls ++TxDtls +++RmtInf ++++Strd +++++RfrdDocInf +++++Nb	M	String(35x)	Invoice numer
	+NtryDtls ++TxDtls +++RmtInf ++++Strd +++++RfrdDocAmt +++++TaxAmt	M	Number 10n,2n	VAT amount
	+NtryDtls ++TxDtls +++RmtInf ++++Strd +++++Invc +++++Id +++++OrgId +++++Othr +++++Id	M	String(14x)	ID numer (TAX identification numer)
	+NtryDtls ++TxDtls +++RmtInf ++++Strd +++++AddtlRmtInf	O	String(33x)	Additional information

Example:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <GetAccountHistoryResponse xmlns="http://www.bankmillennium.pl/CorporateServices">
      <GetAccountHistoryResult>
        <BkToCstmrAcctRpt xmlns="urn:iso:std:iso:20022:tech:xsd:camt.052.001.02">
          <GrpHdr>
            <MsgId>Message ID</MsgId>
            <CreDtTm>2017-09-21T15:49:18.3092286+02:00</CreDtTm>

```

```

</GrpHdr>
<Rpt>
  <Id>Message ID</Id>
  <CreDtTm>2017-09-21T15:49:18.3092286+02:00</CreDtTm>
  <FrToDt>
    <FrDtTm>Date from</FrDtTm>
    <ToDtTm>Date to</ToDtTm>
  </FrToDt>
  <Acct>
    <Id>
      <IBAN> Bank account number </IBAN>
    </Id>
  </Acct>
  <Ntry>
    <Amt Ccy="PLN">Amount</Amt>
    <CdtDbtInd>CRDT</CdtDbtInd>
    <Sts>BOOK</Sts>
    <BookgDt>
      <DtTm>2017-06-26T00:00:00</DtTm>
    </BookgDt>
    <ValDt>
      <DtTm>2017-06-26T00:00:00</DtTm>
    </ValDt>
    <BkTxCd>
      <Domn>
        <Cd>TFRMILWINI</Cd>
        <Fmly>
          <Cd>TFRMILWINI</Cd>
          <SubFmlyCd>Transaction type</SubFmlyCd>
        </Fmly>
      </Domn>
    </BkTxCd>
    <NtryDtls>
      <TxDtls>
        <Refs>
          <Msgld>Transaction number during the day</Msgld>
        </Refs>
        <RltdPties>
          <Dbtr>
            <Nm> Contractor name</Nm>
          </Dbtr>
          <DbtrAcct>
            <Id>
              <Othr>
                <Id> Bank account number </Id>

```

```

        </Othr>
    </Id>
    </DbtrAcct>
</RltdPties>
    <RmtInf>
        <Ustrd>Title of the order</Ustrd>
    </RmtInf>
</TxDtls>
</NtryDtls>
</Ntry>
</Rpt>
</BkToCstmrAcctRpt>
</GetAccountHistoryResult>
</GetAccountHistoryResponse>
</s:Body>
</s:Envelope>

```

An example of a part of response with pagination:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <GetAccountHistoryResponse xmlns="http://www.bankmillennium.pl/CorporateServices">
      <GetAccountHistoryResult>
        <BkToCstmrAcctRpt xmlns="urn:iso:std:iso:20022:tech:xsd:camt.052.001.02">
          <GrpHdr>
            <MsgId>Message ID</MsgId>
            <CreDtTm>2017-11-10T12:02:15.9720051+01:00</CreDtTm>
            <MsgPgntn>
              <PgNb>-History pagination - number</PgNb>
              <LastPgInd>false</LastPgInd>
            </MsgPgntn>
          </GrpHdr>
          <Rpt>
            <Id>Message ID</Id>
            <CreDtTm>2017-11-10T12:02:15.9720051+01:00</CreDtTm>
            <FrToDt>
              <FrDtTm>2015-01-14T00:00:00</FrDtTm>
              <ToDtTm>2017-11-14T00:00:00</ToDtTm>
            </FrToDt>
            <Acct>
              <Id>
                <IBAN>Bank account number</IBAN>
              </Id>
            </Acct>
          </Rpt>
        </BkToCstmrAcctRpt>
      </GetAccountHistoryResult>
    </GetAccountHistoryResponse>
  </s:Body>
</s:Envelope>

```

```

<Amt Ccy="PLN">Amount</Amt>
<CdtDbtInd>DBIT</CdtDbtInd>
<Sts>BOOK</Sts>
<BookgDt>
  <DtTm>2016-11-14T00:00:00</DtTm>
</BookgDt>
<ValDt>
  <DtTm>2016-11-14T00:00:00</DtTm>
</ValDt>
<BkTxCd>
  <Domn>
    <Cd>TFROUT</Cd>
    <Fmly>
      <Cd>TFROUT</Cd>
      <SubFmlyCd>Transaction type</SubFmlyCd>
    </Fmly>
  </Domn>
</BkTxCd>
<NtryDtls>
  <TxDtls>
    <Refs>
      <Msgld>Number of transaction during the day</Msgld>
    </Refs>
    <RltdPties>
      <Cdtr>
        <Nm>Name of beneficiary</Nm>
      </Cdtr>
      <CdtrAcct>
        <Id>
          <Othr>
            <Id> Bank account number </Id>
          </Othr>
        </Id>
      </CdtrAcct>
    </RltdPties>
    <RmtInf>
      <Ustrd> Title of the order </Ustrd>
    </RmtInf>
  </TxDtls>
</NtryDtls>
</Ntry>

```

List of standard transaction codes returned by the banking system

REVDEB	ANULOWANIE OBCIĄŻENIA
REVCRD	ANULOWANIE UZNANIA
DEBPOSTED	OBCIĄŻENIE
TXNDCTDEB	POLECENIE ZAPŁATY
FEEOTH	PROWIZJA
TFRFRGIN	PRZELEW ZAGRANICZNY
REVFRGOUT	STORNO-PRZELEW ZAGR.
TFRFRGOUT	PRZELEW ZAGRANICZNY
TFROUT	PRZELEW DO INNEGO BANKU
TFRTAXOUT	PRZELEW DO URZEDU SKARBOWEGO
TFRBBGWIN	PRZELEW MIĘDZYODDZIAŁOWY
TFRSOROUT	PRZELEW SORBNET
TFRMILWIN	PRZELEW WEWNĄTRZ BANKU
TFRZUSOUT	PRZELEW DO ZUS
CHQCCT	SKUP/INKASO CZEKU
TXNRPYCRD	SPLATA KARTY
TFRSOWIN	STAŁE ZLECENIE WEWNĄTRZ BANKU
TFRSOOUT	STAŁE ZLECENIE ZEWNĘTRZNE
TXNATMPOS	TRANSAKCJA KARTĄ PŁATNICZĄ
CRDRCV	UZNANIE
CRDRCVEXT	UZNANIE OTRZYMANE
CRDRCVEXR	PRZELEW PRZYCHODZĄCY
CRDRCVSOR	UZNANIE OTRZYMANE SORBNET
CRDRCVSBR	UZNANIE OTRZYMANE SYBIR
DEPCSH	WPLATA NA RACHUNEK
CASHWITHDR	WYPŁATA GOTÓWKOWA
TXNATMWTD	WYPŁATA KARTĄ Z BANKOMATU
FEEOTH	PROWIZJA-SORBNET
TXNERLCREP	WCZEŚ.SPLATA KARTY
TFRMILWINI	PRZELEW WEWNĘTRZNY PRZYCHODZĄCY
TFRMILWINO	PRZELEW WEWNĘTRZNY WYCHODZĄCY
SEPAIN	PRZEKAZ SEPA
SEPAOUT	PRZEKAZ SEPA
TXNIPAYOUT	PŁATNOŚĆ INTERNETOWA WYCHODZĄCA
TXNIPAYIN	PŁATNOŚĆ INTERNETOWA PRZYCHODZĄCA
SRPNOUT	PRZELEW NATYCHMIASTOWY WYCHODZĄCY

SRPNIN	PRZELEW NATYCHMIASTOWY PRZYCHODZĄCY
TMNATAOPR	OPERACJE NA LOKATACH
LNNATAOPR	OPERACJE NA KREDYTACH
TXCASHADV	OPERACJE NA KARTACH
TMNASOOPR	OPERACJE SO NA LOKATACH
LNNASOOPR	OPERACJE SO NA KREDYTACH
FEETRIN	OPLATA
TXNECW	Elektron. wypłata got. ECW
FE1	Prow. złożenie zleceń. ECW
FE2	Prowizja za wypłat. ECW
FE3	Prow. nieodebr. wypł. ECW.
CLOSWITHDR	WYPŁATA ZAMKNIĘTA
TXNDCTDEB	ODSETKI ZWROT POL.ZAPŁ.
POSCASHWDR	Wypłata w punkcie akceptującym
CARDTRNX	TRANSAKCJA KARTĄ
SRPNOUTZUS	PRZEL.NATYCH.WYCH.ZUS
SRPNOUTTAX	PRZEL.NATYCH.WYCH.PODATKOWY
SRPNINTAX	PRZEL.NATYCH.PRZYCH.PODATKOWY
TXNCMW5	WYPŁATA OTWARTA
TXNCMW6	WYPŁATA ZAMKNIĘTA W ODDZIALE
TXNCMW7	WYPŁATA ZAMKNIĘTA U KLIENTA
TXNCMW8	WYPŁATA ZAMKNIĘTA W COG
CRDTRTY_21	AUTORYZACJA TRANSAKCJI KARTĄ
CRDTRTY_XX	TRANSAKCJA KARTĄ
DEPCSH	KOREKTA WPLATY NA RACHUNEK
TFRPO	PRZEKAZ POCZTOWY
MILSPLITIN	PRZELEW PRZYCHODZĄCY VAT (wewnątrz Banku)
SPLITIN	PRZELEW PRZYCHODZĄCY VAT
MILSPLITPU	PRZELEW WYCHODZĄCY VAT (wewnątrz Banku)
SPLITOUT	PRZELEW WYCHODZĄCY VAT
TFRVATIN	PRZEKSIEGOWANIE VAT
TFRVATOUT	PRZEKSIEGOWANIE VAT
OWNSPLITIN	PRZELEW WEWNĘTRZNY VAT (transakcja przychodząca)
OWNSPLITOU	PRZELEW WEWNĘTRZNY VAT (transakcja wychodząca)
DCTDEBVAT	PRZELEW POLECENIE ZAPŁATY VAT
DCTDEBVATR	ZWROT POLECENIE ZAPŁATY VAT

3.1.3. GetRecentAccountHistory

The GetRecentAccountHistory method supports the exchange of information between the Customer and the Bank regarding the history of operations on the Customer's account.

It differs from GetAccountHistory in that the history of operations on a given account is always collected from the last call of this service for a given account, thus the Client does not complete the range of dates.

The moment from which the method starts to store data is the date when the virtual user was created.

3.1.3.1. GetRecentAccountHistoryRequest

Sent by the Customer to the Bank. It contains the account number for which the Client expects history.

Message structure:

Field name	XML element name	Status	Format	Comments
Message ID	GetAcctRpt +MsgId ++Id	O	String(30)	Unique for a given recipient. Can not contain spaces.
History pagination	GetAcctRpt +GrpHdr ++MsgPgntn +++PgNb	C	String	If the history search period has more than 600 records, the account history page number is received in the GetAccountHistory response, always entered in the second query in the given download queue
Account number	GetAcctRpt +AcctRptQryDef ++AcctRptCrit +++NewCrit ++++SchCrit +++++AcctId ++++++EQ +++++++IBAN	M	2!a26!n	Account number in IBAN format

Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://www.bankmillennium.pl/CorporateServices"
xmlns:urn="http://www.bankmillennium.pl/CorporateServices">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:GetRecentAccountHistory>
      <tem:request>
        <urn:GetAcctRpt>
          <urn:MsgId>
            <urn:Id>Message ID</urn:Id>
          </urn:MsgId>
```

```

    <urn:AcctRptQryDef>
      <urn:AcctRptCrit>
        <urn:NewCrit>
          <urn:SchCrit>
            <urn:AcctId>
              <urn:EQ>
                <urn:IBAN> Bank account number </urn:IBAN>
              </urn:EQ>
            </urn:AcctId>
          </urn:SchCrit>
        </urn:NewCrit>
      </urn:AcctRptCrit>
    </urn:AcctRptQryDef>
  </urn:GetAcctRpt>
</tem:request>
</tem:GetRecentAccountHistory>
</soapenv:Body>
</soapenv:Envelope>

```

Example with pagination:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:tem="http://www.bankmillennium.pl/CorporateServices"
  xmlns:urn="http://www.bankmillennium.pl/CorporateServices">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:GetRecentAccountHistory>
      <tem:request>
        <urn:GetAcctRpt>
          <urn:MsgId>
            <urn:Id>Message ID</urn:Id>
          </urn:MsgId>
          <urn:MsgPgntn>
            <urn:PgNb>-History pagination - number</urn:PgNb>
          </urn:MsgPgntn>
          <urn:AcctRptQryDef>
            <urn:AcctRptCrit>
              <urn:NewCrit>
                <urn:SchCrit>
                  <urn:AcctId>
                    <urn:EQ>
                      <urn:IBAN> Bank account number </urn:IBAN>
                    </urn:EQ>
                  </urn:AcctId>
                </urn:SchCrit>
              </urn:NewCrit>
            </urn:AcctRptCrit>
          </urn:AcctRptQryDef>
        </urn:GetAcctRpt>
      </tem:request>
    </tem:GetRecentAccountHistory>
  </soapenv:Body>
</soapenv:Envelope>

```

```

        </urn:AcctRptCrit>
    </urn:AcctRptQryDef>
</urn:GetAcctRpt>
</tem:request>
</tem:GetRecentAccountHistory>
</soapenv:Body>
</soapenv:Envelope>

```

3.1.3.2. GetRecentAccountHistoryResponse

Sent by the Bank to the Customer. The structure does not differ from GetAccountHistory. Includes incremental operations since the last GetRecentAccountHistoryRequest call for a given account.

Message structure:

Nazwa pola	Nazwa elementu XML	Status	Format	Uwagi
Message ID	BkToCstmrAcctRpt +GrpHdr ++MsgId	O	String(30)	Unique for a given recipient. Can not contain spaces
Date and time of creation	BkToCstmrAcctRpt +GrpHdr ++CreDtTm	M	ISO DateAndTime	ISO DateAndTime
History pagination	BkToCstmrAcctRpt +GrpHdr ++MsgPgntn	C	String	The unique history pagination number, made available when 600 records are exceeded in the requested time period, the number for pasting to subsequent inquiries for the requested account history
History pagination - the last page	BkToCstmrAcctRpt +GrpHdr ++MsgPgntn	M	String	It shows whether this is the last answer to the desired account history. Possible answers are: 'False' - means that after the query the next pagination number will be sent 'True' - last on the history page
Message ID	BkToCstmrAcctRpt +Rpt ++Id	M	String(30)	The same ID as in the header
Date and time of creation	BkToCstmrAcctRpt +Rpt ++CreDtTm	M	ISO DateAndTime	The same date as in the header
Starting date and time for history	BkToCstmrAcctRpt +Rpt ++FrToDt +++FrDtTm	M	ISO DateAndTime	Only the date is significant, the time is always set to 00:00

End date and time for history	BkToCstmrAcctRpt +Rpt ++FrToDt +++ToDtTm	M		The value is set for <ToDtTm xsi:nil="true"/>
Account number	BkToCstmrAcctRpt +Rpt ++Acct +++Id ++++IBAN	M	2!a26!n	Account number in IBAN format
Item in history	BkToCstmrAcctRpt +Rpt ++Ntry	O	-	It can occur 0 or many times
Fields in the history item (inside the <Ntry> tag):				
Item amount	+Amt	M	String(3)+Nu mber(*,2)	The amount and currency of the operation. Mandatory entering also the currency code (ISO).
Item indicator	+CdtDbtInd	M	String(4)	Acceptable values 'CRDT' - for the amount ≥ 0 'DBIT' - for the amount <0
Item status	+Sts	M	String(4)	Acceptable value BOOK.
Accounting date	+BookgDt ++DtTm	M	ISO DateAndTim e	If there is no information about the accounting time, it will be returned in the form of 00:00
Operation date	+ValDt ++DtTm	M	ISO DateAndTim e	If there is no information about the operation time, it will be returned in the form of 00:00
Banking operation code	+BkTxCd ++Domn +++Cd	M	String	Standard operating code returned by the banking system (list of basic codes is included below)
Extended operation code	+BkTxCd ++Domn +++Fmly ++++Cd	O	String	
Extended operation subcode	+BkTxCd ++Domn +++Fmly ++++SubFmlyCd	O	String	

Transaction number during the day	+NtryDtls ++TxDtls +++Refs ++++MsgId	O	String	Number for downloading incremental history
Contractor	+NtryDtls ++TxDtls +++RltdPties ++++Cdtr	O	-	For debit operations (details of the marker included in the next table)
Name of the contractor	+NtryDtls ++TxDtls +++RltdPties ++++Cdtr ++++Nm	O	String	Contractor's details (Name, Address, city, zip code)
Contractor's account	+NtryDtls ++TxDtls +++RltdPties ++++CdtrAcct ++++Id +++++Othr +++++Id	O	String(30)	For debit operations The account number (regardless of whether IBAN or not) is returned in the element Account is not shown in Balancing VAT transactions
Transfer title - unstructured	+NtryDtls ++TxDtls +++RmtInf ++++Ustrd	O	String(140)	Operation title (unstructured)
Transfer title - structured	+NtryDtls ++TxDtls +++RmtInf ++++Strd +++++RfrdDocInf +++++Nb	M	String(35x)	Invoice numer
	+NtryDtls ++TxDtls +++RmtInf ++++Strd +++++RfrdDocAmt +++++TaxAmt	M	Number 10n,2n	VAT amount
	+NtryDtls ++TxDtls +++RmtInf ++++Strd +++++Invcr +++++Id +++++OrgId +++++Othr +++++Id	M	String(14x)	ID numer (TAX identification numer)

	+NtryDtls ++TxDtls +++RmtInf ++++Strd +++++AddtlRmtInf	O	String(33x)	Additional information
--	--	---	-------------	------------------------

Example:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <GetRecentAccountHistoryResponse xmlns="http://www.bankmillennium.pl/CorporateServices">
      <GetRecentAccountHistoryResult>
        <BkToCstmrAcctRpt xmlns="urn:iso:std:iso:20022:tech:xsd:camt.052.001.02">
          <GrpHdr>
            <MsgId>Message ID</MsgId>
            <CreDtTm>2017-09-21T16:41:45.2559185+02:00</CreDtTm>
          </GrpHdr>
          <Rpt>
            <Id>Message ID</Id>
            <CreDtTm>2017-09-21T16:41:45.2559185+02:00</CreDtTm>
            <FrToDt>
              <FrDtTm>2017-08-31T10:09:13.9839581Z</FrDtTm>
              <ToDtTm xsi:nil="true"/>
            </FrToDt>
            <Acct>
              <Id>
                <IBAN> Bank account number </IBAN>
              </Id>
            </Acct>
            <Ntry>
              <Amt Ccy="PLN">Amount</Amt>
              <CdtDbtInd>CRDT</CdtDbtInd>
              <Sts>BOOK</Sts>
              <BookgDt>
                <DtTm>2017-09-19T00:00:00</DtTm>
              </BookgDt>
              <ValDt>
                <DtTm>2017-09-19T00:00:00</DtTm>
              </ValDt>
              <BkTxCd>
                <Domn>
                  <Cd>TFRSOWIN</Cd>
                  <Fmly>
                    <Cd>TFRSOWIN</Cd>
                    <SubFmlyCd/>

```

```

        </Fmly>
    </Domn>
</BkTxCd>
<NtryDtls>
    <TxDtls>
        <Refs>
            <Msgld>Number of transaction during the day</Msgld>
        </Refs>
        <RltdPties>
            <Dbtr>
                <Nm/>
            </Dbtr>
            <DbtrAcct>
                <Id>
                    <Othr>
                        <Id> Bank account number </Id>
                    </Othr>
                </Id>
            </DbtrAcct>
        </RltdPties>
        <RmtInf>
            <Ustrd> Title of the order </Ustrd>
        </RmtInf>
    </TxDtls>
</NtryDtls>
</Ntry>
</Rpt>
</BkToCstmrAcctRpt>
</GetRecentAccountHistoryResult>
</GetRecentAccountHistoryResponse>
</s:Body>
</s:Envelope>

```

Example with pagination:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <GetRecentAccountHistoryResponse xmlns="http://www.bankmillennium.pl/CorporateServices">
      <GetRecentAccountHistoryResult>
        <BkToCstmrAcctRpt xmlns="urn:iso:std:iso:20022:tech:xsd:camt.052.001.02">
          <GrpHdr>
            <Msgld>Message ID</Msgld>
            <CreDtTm>2017-11-20T16:15:32.2579272+01:00</CreDtTm>
            <MsgPgntn>
              <PgNb>-History pagination - number</PgNb>
            </MsgPgntn>
          </GrpHdr>
        </BkToCstmrAcctRpt>
      </GetRecentAccountHistoryResult>
    </GetRecentAccountHistoryResponse>
  </s:Body>
</s:Envelope>

```

```

        <LastPgInd>false</LastPgInd>
    </MsgPgntn>
</GrpHdr>
<Rpt>
    <Id>Message ID</Id>
    <CreDtTm>2017-11-20T16:15:32.2579272+01:00</CreDtTm>
    <FrToDt>
        <FrDtTm>2017-11-20T13:58:27.7705673Z</FrDtTm>
        <ToDtTm xsi:nil="true"/>
    </FrToDt>
    <Acct>
        <Id>
            <IBAN> Bank account number </IBAN>
        </Id>
    </Acct>

```

3.2. Statements

3.2.1. GetStatement

The GetStatement method supports the exchange of information between the Customer and the Bank regarding statements from the Customer's accounts

3.2.1.1. GetStatementRequest

Sent by the Customer to the Bank. It contains parameters allowing to identify the account, the time when the statements were generated and its format (pdf or MT940) and type (daily, weekly, bi-weekly or monthly).

Message structure:

Field name	XML element name	Status	Format	Comments
Message ID	GetStmnt +MsgId ++Id	M	String(30)	Unique for a given recipient. Can not contain spaces.
Account number	GetStmnt +StmntQryDef> ++StmntCrit +++NewCrit ++++SchCrit +++++AcctId ++++++EQ +++++++IBAN	M	String(30)	Account number in IBAN format It is also possible to use an account number in the BBAN format - the IBAN tag should be changed to BBAN.
Starting date for the history	GetStmnt +StmntQryDef> ++StmntCrit +++NewCrit ++++SchCrit +++++StmntValDt	M	ISO Date	

	+++++DtSch> +++++FrDt			
End date for the history	GetStmt +StmtQryDef> ++StmtCrit +++NewCrit ++++SchCrit +++++StmtValDt +++++DtSch> +++++ToDt	M	ISO DATE	
Format	GetStmt +StmtQryDef ++StmtCrit +++NewCrit ++++SchCrit +++++StmtFrmt	M	String	Specifies the format of the statement: PDF - statement in PDF format MT940 - statement in a text file compliant with the Millenet for Enterprises specification
Statement type	GetStmt +StmtQryDef ++StmtCrit +++NewCrit ++++SchCrit +++++StmtType	M	String	Statement type: Daily / Weekly / BiWeekly / Monthly.

Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://www.bankmillennium.pl/CorporateServices"
xmlns:urn="http://www.bankmillennium.pl/CorporateServices">
```

```
<soapenv:Header/>
```

```
<soapenv:Body>
```

```
<tem:GetStatement>
```

```
<tem:request>
```

```
<urn:GetStmt>
```

```
<urn:MsgId>
```

```
<urn:Id>Message ID</urn:Id>
```

```
</urn:MsgId>
```

```
<urn:StmtQryDef>
```

```
<urn:StmtCrit>
```

```
<urn:NewCrit>
```

```
<urn:SchCrit>
```

```
<urn:AcctId>
```

```
<urn:EQ>
```

```
<urn:IBAN> Bank account number </urn:IBAN>
```

```
</urn:EQ>
```

```
</urn:AcctId>
```

```
<urn:StmtValDt>
```

```
<urn:DtSch>
```

```

        <urn:FrDt>Date from</urn:FrDt>
        <urn:ToDt>Date to</urn:ToDt>
    </urn:DtSch>
</urn:StmtValDt>
    <urn:StmtFrmt>Statement format</urn:StmtFrmt>
    <urn:StmtType>Statement frequency</urn:StmtType>
</urn:SchCrit>
</urn:NewCrit>
</urn:StmtCrit>
</urn:StmtQryDef>
</urn:GetStmt>
</tem:request>
</tem:GetStatement>
</soapenv:Body>
</soapenv:Envelope>

```

3.2.1.2. GetStatementResponse

Sent by the Bank to the Customer. It will contain extracts of a given type in the form of Base64Binary files in the format expected by the Customer.

Message structure:

Field name	XML element name	Status	Format	Comments
Message ID	GetStatementResult +MsgId ++Id	0	String(30)	Unique for a given recipient. Can not contain spaces. Message ID. The value of this field is always the same as in the query.
File name	GetStatementResult +RptDtIs ++DocumentDetails +++RptNm		String	
Format	GetStatementResult +RptDtIs ++DocumentDetails +++RptType		String	
Size	GetStatementResult +RptDtIs ++DocumentDetails +++RptSize		Number	
Statement file	GetStatementResult +RptDtIs ++RptFile		Base64Binary	The content of the file is attached to the SOAP message in the base64 format

Example:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <GetStatementResponse xmlns="http://www.bankmillennium.pl/CorporateServices">
      <GetStatementResult>
        <Rpt xmlns="GetStatementResponse">
          <MsgId>
            <Id>Message ID</Id>
          </MsgId>
          <RptDtls>
            <DocumentDetails>
              <RptNm>Statement name</RptNm>
              <RptType>Statement format</RptType>
              <RptSize>File size</RptSize>
              <RptFile>Statement file</RptFile>
            </DocumentDetails>
          </RptDtls>
        </Rpt>
      </GetStatementResult>
    </GetStatementResponse>
  </s:Body>
</s:Envelope>

```

3.2.2. GetRecentStatement

The GetRecentStatement method supports the exchange of information between the Customer and the Bank regarding statements from the Customer's accounts.

It differs from GetStatement in that always statements from the Customer's account are taken from the last call of this service for a given account, thus the Client does not complete the range of dates.

The moment from which the method starts to store data is the date when the virtual user was created.

3.2.2.1. GetRecentStatementRequest

Sent by the Customer to the Bank. It contains the invoice number, format and type of the statement.

Message structure:

Field name	XML element name	Status	Format	Comments
Message ID	GetStmt +MsgId ++Id	O	String(30)	Unique for a given recipient. Can not contain spaces.
Account number	GetStmt +StmtQryDef> ++StmtCrit +++NewCrit	M	String(30)	Account number in IBAN format It is also possible to use an account number in the BBAN

	++++SchCrit +++++AcctId ++++++EQ +++++++IBAN			format - the IBAN tag should be changed to BBAN.
Format	GetStmt +StmtQryDef ++StmtCrit +++NewCrit ++++SchCrit +++++StmtFrmt		String	Specifies the format of the statement: PDF - statement in PDF format MT940 - statement in a text file compliant with the Millenet for Enterprises specification
Statement type	GetStmt +StmtQryDef ++StmtCrit +++NewCrit ++++SchCrit +++++StmtType		String	Statement type: Daily / Weekly / BiWeekly / Monthly.

Example:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://www.bankmillennium.pl/CorporateServices"
xmlns:urn="http://www.bankmillennium.pl/CorporateServices">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:GetRecentStatement>
      <tem:request>
        <urn:GetStmt>
          <urn:MsgId>
            <urn:Id>Message ID</urn:Id>
          </urn:MsgId>
          <urn:StmtQryDef>
            <urn:StmtCrit>
              <urn:NewCrit>
                <urn:SchCrit>
                  <urn:AcctId>
                    <urn:EQ>
                      <urn:IBAN> Bank account number </urn:IBAN>
                    </urn:EQ>
                  </urn:AcctId>
                <urn:StmtFrmt>Statement format</urn:StmtFrmt>
                <urn:StmtType>Statement frequency</urn:StmtType>
              </urn:SchCrit>
            </urn:NewCrit>
          </urn:StmtCrit>
        </urn:StmtQryDef>
      </urn:GetStmt>
    </tem:request>
  </soapenv:Body>
</soapenv:Envelope>

```

```

    </tem:GetRecentStatement>
  </soapenv:Body>
</soapenv:Envelope>

```

3.2.2.2. GetRecentStatementResponse

Sent by the Bank to the Customer. The structure does not differ from GetStatementResponse. Includes incremental statements since the last GetRecentStatementResponse call for a given account and type of statement.

Message structure:

Field name	XML element name	Status	Format	Comments
Message ID	GetStatementResult +MsgId ++Id	0	String(30)	Unique for a given recipient. Can not contain spaces. Message ID. The value of this field is always the same as in the query.
File name	GetStatementResult +RptDtls ++DocumentDetails +++RptNm		String	
Format	GetStatementResult +RptDtls ++DocumentDetails +++RptType		String	
Size	GetStatementResult +RptDtls ++DocumentDetails +++RptSize		Number	
Statement file	GetStatementResult +RptDtls ++RptFile		Base64Binary	The content of the file is attached to the SOAP message in the base64 format.

Example:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <GetStatementResponse xmlns="http://www.bankmillennium.pl/CorporateServices">
      <GetStatementResult>
        <Rpt xmlns="GetStatementResponse">
          <MsgId>
            <Id>Message ID</Id>
          </MsgId>
          <RptDtls>
            <DocumentDetails>
              <RptNm>Statement name</RptNm>
            </DocumentDetails>
          </RptDtls>
        </Rpt>
      </GetStatementResult>
    </GetStatementResponse>
  </s:Body>
</s:Envelope>

```

```

        <RptType>Statement format</RptType>
        <RptSize>Statement size</RptSize>
        <RptFile>Statement file</RptFile>
    </DocumentDetails>
</RptDtIs>
</Rpt>
</GetStatementResult>
</GetStatementResponse>
</s:Body>
</s:Envelope>

```

3.3. Payments

3.3.1. CreditTransfer

The method allows to send payment orders for the transaction types described below. One CreditTransfer message may contain different types of transactions.

Message structure

Message header

Field name	XML element name	Status	Format	Comments
Message ID	CstmrCdtTrfInitn +GrpHdr ++MsgId	O	35x	Unique for a given message. Can not contain spaces. Message ID.
Time of creation	+GrpHdr ++CreDtTm	M	RRRR-MM-DDThh:mm:ss	2016-05-05T11:00:00
Number of orders in message	+GrpHdr ++NbOfTxS	M	10d	
Name of the orderer	+GrpHdr ++InitgPty +++Nm	M	140x	e.g. user of the FK system

Distinguish between types of orders

Types of orders	CstmrCdtTrfInitn/PmtInf/CdtTrfTxInf/		
	Purp/Prtry	PmtTpInf/CtgyPurp/Cd	PmtTpInf/SvcLvl/Cd
Domestic standard in PLN Domestic standard VAT in PLN	PLKR SPKR	-	STAN - standard transfer
Domestic express elixir Domestic express elixir VAT	PLKR SPKR	-	SRPN - Express Elixir
Domestic sorbnet Domestic sorbnet VAT	PLKR SPKR	-	RTGS - SORBNET
TAX	TAXS	-	-

Social	-	SSBE	-
Postal order	POST	-	-
From prepaid card	PREP	-	-
Internal - between Client's settlement accounts - on charge card - on debit card - between Client's VAT accounts	INTS or INTC or INTP or INTV	-	-
Foreign Domestic in currency different than PLN	FRGN	-	-

Creating groups of orders - Payment Information

The rules for creating packages:

- one message gives the possibility to save orders to many folders
- the name of the package is defined in the field <PmtInfId> and by repeated repetition of Payment Information blocks, the orders placed in them will be combined into packages whose names have been specified
- each package can contain orders of various types

Field name	XML element name	Status	Format	Comments
Folder name	+PmtInf ++PmtInfId	M	45x	
Type of operation	+PmtInf ++PmtMtd	M	3a!	Constant value „TRF“
Date of execution	+PmtInf ++ReqdExctnDt	M	RRRR-MM-DD	
Name of orderer	+PmtInf ++Dbtr +++Nm	C	78x	Required if option „override orderer details“ is enabled
Country symbol	+PmtInf ++Dbtr +++PstlAdr ++++Ctry	O	2a!	Country code according to ISO 31666
Address of orderer	+PmtInf ++Dbtr +++PstlAdr ++++AdrLine	O	35x	
Account number of orderer	+PmtInf ++DbtrAcct +++Id ++++IBAN	M	2!a26!n/26!n	In IBAN or NRB format

	lub +PmtInf ++DbtrAcct +++Id ++++Othr +++++Id			
Constant value "PLKNR"	+PmtInf ++DbtrAgt +++ FinInstnId ++++ClrSysMmbld +++++ClrSysId +++++Cd	M	26!n	
Orderer bank identifier	+PmtInf ++DbtrAgt +++FinInstnId ++++ClrSysMmbld +++++Mmbld	M	8x	Bank's settlement number
Account for fees	+PmtInf ++ChrgsAcct +++Id ++++IBAN lub +PmtInf ++ChrgsAcct +++Id ++++Othr +++++Id	O	2!a26!n/26!n	In IBAN or NRB format used for foreign transfers

3.3.1.1. CreditTransferRequest

3.3.1.1.1. Domestic

Field name	XML element name	Status	Format	Comments
Message ID	+PmtInf ++CdtTrfTxInf +++PmtId ++++EndToEndId	M	16x	
Execution channel	+PmtInf ++CdtTrfTxInf +++PmtTpInf ++++SvcLvl +++++Cd	M	4a!	Constant value „STAN"

Order amount	+PmtInf ++CdtTrfTxInf +++Amt ++++InstdAmt	M	15d	Gross amount in terms of VAT transfer
Order currency	+PmtInf ++CdtTrfTxInf +++Amt ++++InstdAmt@Ccy	M	3a!	Constant value = PLN in terms of VAT transfer
Beneficiary's bank ID	+PmtInf ++CdtTrfTxInf +++CdtrAgt ++++FinInstnId +++++ClrSysMmbld +++++Mmbld	M	8x	
Name of the beneficiary	+PmtInf ++CdtTrfTxInf +++Cdtr ++++Nm	M	70x	
Country	+PmtInf ++CdtTrfTxInf +++Cdtr ++++PstlAdr +++++Ctry	O	2a!	
Address of the beneficiary	+PmtInf ++CdtTrfTxInf +++Cdtr +++PstlAdr ++++AdrLine	O	70x	
Beneficiary's bank account number	+PmtInf ++CdtTrfTxInf +++CdtrAcct ++++Id +++++IBAN lub +PmtInf ++CdtTrfTxInf +++CdtrAcct ++++Id +++++Othr +++++Id	M	2!a26!n/26!n	In IBAN lub NRB format
Order type	+PmtInf ++CdtTrfTxInf +++Purp	M	4a!	Constant value "PLKR" or "SPKR" in terms of VAT transfer

	++++Prtry				
Title of the order	+PmtInf ++CdtTrfTxInf +++RmtInf ++++Ustrd	M	140x		
Order title for VAT transfer					
Order title - unstructured	+PmtInf ++CdtTrfTxInf +++RmtInf ++++Ustrd	M	140x	It can not occur if the title field of the structured order has been filled out The field can only appear once. For VAT transfers, a field built using code words: /VAT/10n,2n/IDC/14x/INV/35x/TXT/33x 1. / VAT / - VAT amount, in the range from 0.01 to 9 999 999 999.99 and not greater than the (Amount - Instructed Amount), (mandatory field), 2. / IDC / - tax identification number of the VAT payer - invoice issuer (mandatory field) **, 3. / INV / - invoice number * (mandatory field), 4. / TXT / - description of inconvenience (optional field). Code words must occur in the order described above.	
Order title - structured	+PmtInf ++CdtTrfTxInf +++RmtInf ++++Strd +++++RfrdDocInf +++++Nb	M	35x	Invoice number *	The order title is structured It can not occur if the unassured order title field has been filled out
	+PmtInf ++CdtTrfTxInf +++RmtInf ++++Strd +++++RfrdDocAmt +++++TaxAmt	M	10,2n	The amount of VAT The VAT amount must be in the range from 0.01 to 9 999 999 999.99 and not greater than the Transfer amount	
	+PmtInf ++CdtTrfTxInf +++RmtInf ++++Strd +++++Invcr	M	14x	Identification number **	

	+++++Id +++++OrgId +++++Othr +++++Id				
	+PmtInf ++CdtTrfTxInf +++RmtInf ++++Strd +++++AddtlRmtInf	0	33x	Additional info / description	
Recipient's tax identification number	+RmtInf ++Strd +++Invcr ++++Id +++++OrgId +++++Other +++++Id	0	10!n	Recipient's tax identification number. Specify if the recipient's account is to be checked on the White List.	

Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
```

```
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.03">
```

```
<soapenv:Header/>
```

```
<soapenv:Body>
```

```
<CstmrCdtTrfInitn>
```

```
<GrpHdr>
```

```
<MsgId>Message ID</MsgId>
```

```
<CreDtTm>2016-12-01T11:00:00</CreDtTm>
```

```
<NbOfTx>Number of transaction in message</NbOfTx>
```

```
<InitgPty>
```

```
<Nm>Orderer name (np. użytkownik FK)</Nm>
```

```
</InitgPty>
```

```
</GrpHdr>
```

```
<PmtInf>
```

```
<PmtInfId>Folder name</PmtInfId>
```

```
<PmtMtd>TRF</PmtMtd>
```

```
<ReqdExctnDt> Execution date </ReqdExctnDt>
```

```
<Dbtr>
```

```
<Nm>Orderer name</Nm>
```

```
<PstlAdr>
```

```
<AdrLine>Address</AdrLine>
```

```
<AdrLine>Address continuation</AdrLine>
```

```
</PstlAdr>
```

```
</Dbtr>
```

```
<DbtrAcct>
```

```
<Id>
```

```

    <Othr>
      <Id> Bank account number </Id>
    </Othr>
  </Id>
</DbtrAcct>
<DbtrAgt>
  <FinInstnId>
    <ClrSysMmbld>
      <ClrSysId>
        <Cd>PLKNR</Cd>
      </ClrSysId>
      <Mmbld>Bank ID</Mmbld>
    </ClrSysMmbld>
  </FinInstnId>
</DbtrAgt>
<CdtTrfTxInf>
  <PmtId>
    <InstrId>Reference</InstrId>
    <EndToEndId>Transaction ID</EndToEndId>
  </PmtId>
  <PmtTpInf>
    <SvcLvl>
      <Cd>STAN</Cd>
    </SvcLvl>
  </PmtTpInf>
  <Amt>
    <InstdAmt Ccy="PLN">Amount</InstdAmt>
  </Amt>
  <CdtrAgt>
    <FinInstnId>
      <ClrSysMmbld>
        <Mmbld>Bank ID</Mmbld>
      </ClrSysMmbld>
    </FinInstnId>
  </CdtrAgt>
  <Cdtr>
    <Nm>Name of beneficiary</Nm>
    <PstlAdr>
      <AdrLine>Address</AdrLine>
      <AdrLine>Address continuation</AdrLine>
    </PstlAdr>
  </Cdtr>
  <CdtrAcct>
    <Id>
      <Othr>

```

```

        <Id> Bank account number </Id>
    </Othr>
</Id>
</CdtrAcct>
<Purp>
    <Prtry>PLKR</Prtry>
</Purp>
<RmtInf>
    <Ustrd> Title of the order </Ustrd>
</RmtInf>
</CdtTrfTxInf>
</PmtInf>
</CstmrCdtTrfInitn>
</soapenv:Body>
</soapenv:Envelope>

```

3.3.1.1.2. Express

Field name	XML element name	Status	Format	Comments
Message ID	+PmtInf ++CdtTrfTxInf +++PmtId ++++EndToEndId	M	16x	
Execution channel	+PmtInf ++CdtTrfTxInf +++PmtTpInf ++++SvcLvl +++++Cd	M	4a!	Constant value „SRPN“
Order amount	+PmtInf ++CdtTrfTxInf +++Amt ++++InstdAmt	M	15d	Gross amount in terms of VAT transfer
Order currency	+PmtInf ++CdtTrfTxInf +++Amt ++++InstdAmt@Ccy	M	3a!	Constant value = PLN in terms of VAT transfer
Beneficiary's bank ID	+PmtInf ++CdtTrfTxInf +++CdtrAgt ++++FinInstnId +++++ClrSysMmbld ++++++Mmbld	M	8x	
Name of the beneficiary	+PmtInf ++CdtTrfTxInf +++Cdtr ++++Nm	M	70x	

Country	+PmtInf ++CdtTrfTxInf +++Cdtr ++++PstlAdr +++++Ctry	O	2a!	
Address of the beneficiary	+PmtInf ++CdtTrfTxInf +++Cdtr +++PstlAdr ++++AdrLine	O	70x	
Beneficiary's bank account number	+PmtInf ++CdtTrfTxInf +++CdtrAcct ++++Id +++++IBAN lub +PmtInf ++CdtTrfTxInf +++CdtrAcct ++++Id +++++Othr +++++Id	M	2!a26!n/26!n	In IBAN lub NRB format
Order type	+PmtInf ++CdtTrfTxInf +++Purp ++++ Prtry	M	4a!	Constant value "PLKR" or "SPKR" in terms of VAT transfer
Title of the order	+PmtInf ++CdtTrfTxInf +++RmtInf ++++Ustrd	M	140x	
Order title for VAT transfer				
Order title - unstructured	+PmtInf ++CdtTrfTxInf +++RmtInf ++++Ustrd	M	140x	It can not occur if the title field of the structured order has been filled out The field can only appear once. For VAT transfers, a field built using code words: /VAT/10n,2n/IDC/14x/INV/35x/TXT/33x 1. / VAT / - VAT amount, in the range from 0.01 to 9 999 999 999.99 and not greater than the (Amount - Instructed Amount), (mandatory field),

				2. / IDC / - tax identification number of the VAT payer - invoice issuer (mandatory field) **, 3. / INV / - invoice number * (mandatory field), 4. / TXT / - description of inconvenience (optional field). Code words must occur in the order described above.	
Order title - structured	+PmtInf ++CdtTrfTxInf +++RmtInf ++++Strd +++++RfrdDocInf ++++++Nb	M	35x	Invoice number *	The order title is structured It can not occur if the unassured order title field has been filled out
	+PmtInf ++CdtTrfTxInf +++RmtInf ++++Strd +++++RfrdDocAmt ++++++TaxAmt	M	10,2n	The amount of VAT The VAT amount must be in the range from 0.01 to 9 999 999 999.99 and not greater than the Transfer amount	
	+PmtInf ++CdtTrfTxInf +++RmtInf ++++Strd +++++Invcr ++++++Id ++++++OrgId ++++++Othr ++++++Id	M	14x	Identification number **	
	+PmtInf ++CdtTrfTxInf +++RmtInf ++++Strd +++++AddtlRmtInf	O	33x	Additional info / description	

Example:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.03">
  <soapenv:Header/>
  <soapenv:Body>
    <CstmrCdtTrfInitn>

```

<GrpHdr>
 <MsgId>Message ID</MsgId>
 <CreDtTm>2016-12-01T11:00:00</CreDtTm>
 <NbOfTxs>Number of transaction in message</NbOfTxs>
 <InitgPty>
 <Nm>Orderer name (np. użytkownik FK)</Nm>
 </InitgPty>
</GrpHdr>
<PmtInf>
 <PmtInfId>Folder name</PmtInfId>
 <PmtMtd>TRF</PmtMtd>
 <ReqdExctnDt> Execution date </ReqdExctnDt>
 <Dbtr>
 <Nm>Orderer name</Nm>
 <PstlAdr>
 <AdrLine>Address</AdrLine>
 <AdrLine>Address continuation</AdrLine>
 </PstlAdr>
 </Dbtr>
 <DbtrAcct>
 <Id>
 <Othr>
 <Id> Bank account number </Id>
 </Othr>
 </Id>
 </DbtrAcct>
 <DbtrAgt>
 <FinInstnId>
 <ClrSysMmbld>
 <ClrSysId>
 <Cd>PLKNR</Cd>
 </ClrSysId>
 <Mmbld>Bank ID</Mmbld>
 </ClrSysMmbld>
 </FinInstnId>
 </DbtrAgt>
 <CdtTrfTxInf>
 <PmtId>
 <EndToEndId>Transaction ID</EndToEndId>
 </PmtId>
 <PmtTplInf>
 <SvcLvl>
 <Cd>SRPN</Cd>
 </SvcLvl>
 </PmtTplInf>

```

<Amt>
  <InstdAmt Ccy="PLN"> Amount </InstdAmt>
</Amt>
<CdtrAgt>
  <FinInstnId>
    <ClrSysMmbld>
      <Mmbld>Bank ID</Mmbld>
    </ClrSysMmbld>
  </FinInstnId>
</CdtrAgt>
<Cdtr>
  <Nm>Beneficiary name</Nm>
  <PstlAdr>
    <Ctry>PL</Ctry>
    <AdrLine>Address</AdrLine>
    <AdrLine>Address continuation</AdrLine>
  </PstlAdr>
</Cdtr>
<CdtrAcct>
  <Id>
    <Othr>
      <Id> Bank account number </Id>
    </Othr>
  </Id>
</CdtrAcct>
<Purp>
  <Prtry>PLKR</Prtry>
</Purp>
<RmtInf>
  <Ustrd> Title of the order </Ustrd>
</RmtInf>
</CdtTrfTxInf>
</PmtInf>
</CstmrCdtTrfInitn>
</soapenv:Body>
</soapenv:Envelope>

```

3.3.1.1.3. Sorbnet

Field name	XML element name	Status	Format	Comments
Message ID	+PmtInf ++CdtTrfTxInf +++PmtId ++++EndToEndId	M	16x	
Execution channel	+PmtInf ++CdtTrfTxInf	M	4a!	Constant value „RTGS“

	+++PmtTpInf ++++SvcLvl +++++Cd			
Order amount	+PmtInf ++CdtTrfTxInf +++Amt ++++InstdAmt	M	15d	Gross amount in terms of VAT transfer
Order currency	+PmtInf ++CdtTrfTxInf +++Amt ++++InstdAmt@Ccy	M	3a!	Constant value = PLN in terms of VAT transfer
Beneficiary's bank ID	+PmtInf ++CdtTrfTxInf +++CdtrAgt ++++FinInstnId +++++ClrSysMmbld +++++Mmbld	M	8x	
Name of the beneficiary	+PmtInf ++CdtTrfTxInf +++Cdtr ++++Nm	M	70x	
Country	+PmtInf ++CdtTrfTxInf +++Cdtr ++++PstlAdr +++++Ctry	O	2a!	
Address of the beneficiary	+PmtInf ++CdtTrfTxInf +++Cdtr +++PstlAdr ++++AdrLine	O	70x	
Beneficiary's bank account number	+PmtInf ++CdtTrfTxInf +++CdtrAcct ++++Id +++++IBAN lub +PmtInf ++CdtTrfTxInf +++CdtrAcct ++++Id +++++Othr +++++Id	M	2!a26!n/26!n	In IBAN lub NRB format

Order type	+PmtInf ++CdtTrfTxInf +++Purp ++++Prtry	M	4a!	Constant value "PLKR" or "SPKR" in terms of VAT transfer	
Title of the order	+PmtInf ++CdtTrfTxInf +++RmtInf ++++Ustrd	M	140x		
Order title for VAT transfer					
Order title - unstructured	+PmtInf ++CdtTrfTxInf +++RmtInf ++++Ustrd	M	140x	It can not occur if the title field of the structured order has been filled out The field can only appear once. For VAT transfers, a field built using code words: /VAT/10n,2n/IDC/14x/INV/35x/TXT/33x 1. / VAT / - VAT amount, in the range from 0.01 to 9 999 999 999.99 and not greater than the (Amount - Instructed Amount), (mandatory field), 2. / IDC / - tax identification number of the VAT payer - invoice issuer (mandatory field) **, 3. / INV / - invoice number * (mandatory field), 4. / TXT / - description of inconvenience (optional field). Code words must occur in the order described above.	
Order title - structured	+PmtInf ++CdtTrfTxInf +++RmtInf ++++Strd +++++RfrdDocInf +++++Nb	M	35x	Invoice number *	The order title is structured
	+PmtInf ++CdtTrfTxInf +++RmtInf ++++Strd +++++RfrdDocAmt +++++TaxAmt	M	10,2n	The amount of VAT The VAT amount must be in the range from 0.01 to 9 999 999 999.99 and not greater than the Transfer amount	
	+PmtInf ++CdtTrfTxInf	M	14x	Identification number **	
It can not occur if the unassured order title field has been filled out					

	+++RmtInf ++++Strd +++++Invcr +++++Id +++++OrgId +++++Othr +++++Id				
	+PmtInf ++CdtTrfTxInf +++RmtInf ++++Strd +++++AddtlRmtInf	0	33x	Additional info / description	

Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
```

```
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.03">
```

```
<soapenv:Header/>
```

```
<soapenv:Body>
```

```
<CstmrCdtTrfInitn>
```

```
<GrpHdr>
```

```
<MsgId>Message ID</MsgId>
```

```
<CreDtTm>2016-12-01T11:00:00</CreDtTm>
```

```
<NbOfTx>Number of transaction in message</NbOfTx>
```

```
<InitgPty>
```

```
<Nm>Orderer name (np. użytkownik FK)</Nm>
```

```
</InitgPty>
```

```
</GrpHdr>
```

```
<PmtInf>
```

```
<PmtInfId>Folder name</PmtInfId>
```

```
<PmtMtd>TRF</PmtMtd>
```

```
<ReqdExctnDt> Execution date </ReqdExctnDt>
```

```
<Dbtr>
```

```
<Nm>Orderer name</Nm>
```

```
<PstlAdr>
```

```
<AdrLine>Address</AdrLine>
```

```
<AdrLine>Address continuation</AdrLine>
```

```
</PstlAdr>
```

```
</Dbtr>
```

```
<DbtrAcct>
```

```
<Id>
```

```
<Othr>
```

```
<Id> Bank account number </Id>
```

```
</Othr>
```

```
</Id>
```

```

</DbtrAcct>
<DbtrAgt>
  <FinInstnId>
    <ClrSysMmbld>
      <ClrSysId>
        <Cd>PLKNR</Cd>
      </ClrSysId>
      <Mmbld>Bank ID</Mmbld>
    </ClrSysMmbld>
  </FinInstnId>
</DbtrAgt>
<CdtTrfTxInf>
  <PmtId>
    <EndToEndId> Transaction ID </EndToEndId>
  </PmtId>
  <PmtTplInf>
    <SvcLvl>
      <Cd>RTGS</Cd>
    </SvcLvl>
  </PmtTplInf>
  <Amt>
    <InstdAmt Ccy="PLN"> Amount </InstdAmt>
  </Amt>
  <CdtrAgt>
    <FinInstnId>
      <ClrSysMmbld>
        <Mmbld> Bank ID </Mmbld>
      </ClrSysMmbld>
    </FinInstnId>
  </CdtrAgt>
  <Cdtr>
    <Nm>Beneficiary name</Nm>
    <PstlAdr>
      <Ctry>Kod kraju</Ctry>
      <AdrLine>Address</AdrLine>
      <AdrLine>Address continuation</AdrLine>
    </PstlAdr>
  </Cdtr>
  <CdtrAcct>
    <Id>
      <Othr>
        <Id> Bank account number </Id>
      </Othr>
    </Id>
  </CdtrAcct>

```

```

    <Purp>
      <Prtry>PLKR</Prtry>
    </Purp>
    <RmtInf>
      <Ustrd> Title of the order </Ustrd>
    </RmtInf>
  </CdtTrfTxInf>
</PmtInf>
</CstmrCdtTrfInitn>
</soapenv:Body>
</soapenv:Envelope>

```

3.3.1.1.4. Internal

Field name	XML element name	Status	Format	Comments
Message ID	+PmtInf ++CdtTrfTxInf +++PmtId ++++EndToEndId	M	16x	
Order amount	+PmtInf ++CdtTrfTxInf +++Amt ++++InstdAmt	M	15d	
Order currency	+PmtInf ++CdtTrfTxInf +++Amt ++++InstdAmt@Ccy	M	3a!	
Beneficiary's bank account number	+PmtInf ++CdtTrfTxInf +++CdtrAcct ++++Id +++++IBAN lub +PmtInf ++CdtTrfTxInf +++CdtrAcct ++++Id +++++Othr +++++Id	M	2!a26!n/26!n	In IBAN lub NRB format In the case of VAT transfer, please provide the account number, which is linked to a different VAT account than the settlement account provided in the account to debit field.
Order type	+PmtInf ++CdtTrfTxInf +++Purp ++++Prtry	M	4a!	Value "INTS" or "INTC" or "INTP" INTS - transfer to settlement account INTC - charge card repayment

				INTP - crediting of prepaid card INTV - transfer beetwen VAT accounts	
Title of the order	+PmtInf ++CdtTrfTxInf +++RmtInf ++++Ustrd	M	140x		
Order title for VAT transfer					
Order title - unstructured	+PmtInf ++CdtTrfTxInf +++RmtInf ++++Ustrd	M	140x	It can not occur if the title field of the structured order has been filled out The field can only appear once. For VAT transfers, a field built using code words: /VAT/10n,2n/IDC/14x/INV/35x/TXT/33x 1. / VAT / - VAT amount, in the range from 0.01 to 9 999 999 999.99 equal to transfer amount, (mandatory field), 2. / IDC / - constant value: company's NIP (mandatory field) **, 3. / INV / - constant value: Przekazanie własne* (mandatory field), 4. / TXT / - description of inconvenience (optional field). Code words must occur in the order described above.	
Order title - structured	+PmtInf ++CdtTrfTxInf +++RmtInf ++++Strd +++++RfrdDocInf +++++Nb	M	35x	Constant value: Przekazanie własne *	The order title is structured It can not occur if the unassured order title field has been filled out
	+PmtInf ++CdtTrfTxInf +++RmtInf ++++Strd +++++RfrdDocAmt +++++TaxAmt	M	10,2n	The amount of VAT The VAT amount must be in the range from 0.01 to 9 999 999 999.99 and not greater than the Transfer amount	
	+PmtInf ++CdtTrfTxInf +++RmtInf ++++Strd	M	14x	Constant value: company's NIP **	

	+++++Invcr +++++Id +++++Orgld +++++Othr +++++Id				
	+PmtInf ++CdtTrfTxInf +++RmtInf ++++Strd +++++AddtlRmtInf	0	33x	Additional info / description	

Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.03">
```

```
<soapenv:Header/>
```

```
<soapenv:Body>
```

```
<CstmrCdtTrfInitn>
```

```
<GrpHdr>
```

```
<MsgId>Message ID</MsgId>
```

```
<CreDtTm>2016-12-01T11:00:00</CreDtTm>
```

```
<NbOfTxs> Number of transaction in message </NbOfTxs>
```

```
<InitgPty>
```

```
<Nm>Orderer name (np. użytkownik FK)</Nm>
```

```
</InitgPty>
```

```
</GrpHdr>
```

```
<PmtInf>
```

```
<PmtInfId>Folder name</PmtInfId>
```

```
<PmtMtd>TRF</PmtMtd>
```

```
<ReqdExctnDt> Execution date </ReqdExctnDt>
```

```
<Dbtr>
```

```
<Nm>Orderer name</Nm>
```

```
<PstlAdr>
```

```
<AdrLine>Address</AdrLine>
```

```
<AdrLine>Address continuation</AdrLine>
```

```
</PstlAdr>
```

```
</Dbtr>
```

```
<DbtrAcct>
```

```
<Id>
```

```
<Othr>
```

```
<Id> Bank account number </Id>
```

```
</Othr>
```

```
</Id>
```

```
</DbtrAcct>
```

```
<DbtrAgt>
```

```

<FinInstnId>
  <ClrSysMmbld>
    <ClrSysId>
      <Cd>PLKNR</Cd>
    </ClrSysId>
    <Mmbld>Bank ID</Mmbld>
  </ClrSysMmbld>
</FinInstnId>
</DbtrAgt>
<CdtTrfTxInf>
  <PmtId>
    <EndToEndId> Transaction ID </EndToEndId>
  </PmtId>
  <Amt>
    <InstdAmt Ccy="PLN"> Amount </InstdAmt>
  </Amt>
  <CdtrAgt>
    <FinInstnId>
      <ClrSysMmbld>
        <Mmbld> Bank ID </Mmbld>
      </ClrSysMmbld>
    </FinInstnId>
  </CdtrAgt>
  <CdtrAcct>
    <Id>
      <Othr>
        <Id> Bank account number </Id>
      </Othr>
    </Id>
  </CdtrAcct>
  <Purp>
    <Prtry>INTS</Prtry>
  </Purp>
  <RmtInf>
    <Ustrd> Title of the order </Ustrd>
  </RmtInf>
</CdtTrfTxInf>
</PmtInf>
</CstmrCdtTrfInitn>
</soapenv:Body>
</soapenv:Envelope>

```

3.3.1.1.5. Foreign

Field name	XML element name	Status	Format	Comments
Business category	+PmtInf	0	3a!	Required for transfers:

	++CdtTrfTxInf +++PmtId ++++InstrId			- to China and in CNY. Available values: „GOD” - Payment for goods „STR” - Payment for services „CTF” - Capital transfer „OCA” - Other account operations - to United Arab Emirates. Available values: „ATS” - Air Transport „COM” - Commission „COP” - Compensation „EDU” - Education „FIS” - Financial services „GDI” - Payment for goods „OAT” - Own transfer „SAL” - Salaries „STR” - Tourism
Message ID	+PmtInf ++CdtTrfTxInf +++PmtId ++++EndToEndId	M	16x	
Execution mode	+PmtInf ++CdtTrfTxInf +++PmtTpInf ++++InstrPrty	C	4a!	„NORM” - Standard „HIGH” - Urgent „HIGH” - Ekspres (see below)
Execution mode	+PmtInf ++CdtTrfTxInf +++PmtTpInf ++++SvcLvl ++++Cd	M	4a!	For execution mode „Ekspres” value „SDVA”
Order amount	+PmtInf ++CdtTrfTxInf +++Amt ++++InstdAmt	M	15d	
Order currency	+PmtInf ++CdtTrfTxInf +++Amt ++++InstdAmt@Ccy	M	3a!	
Fees and commissions settlement mode	+PmtInf ++CdtTrfTxInf +++ChrgBr	O	4a!	„CRED” - commissions and fees are covered by the beneficiary „DEBT” - commissions and fees are covered by orderer

				„SHAR” - Bank’s commissions and fees are covered by orderer and commissions and fees of other Banks are covered by beneficiary
Beneficiary’s bank SWIFT code	+PmtInf ++CdtTrfTxInf +++CdtrAgt ++++FinInstnId +++++BIC	C	11c for SWIFT or 40c for NID	Beneficiary Bank SWIFT or NID (National Bank Code) code If SWIFT is not known, NID code is required. Phrase “NID:” before the code is required ie: <BIC>NID:100200027</BIC>
The name of the beneficiary's bank	+PmtInf ++CdtTrfTxInf +++CdtrAgt ++++FinInstnId +++++Nm	O	70x	If BIC is given, the name is optional. If no BIC is given, the name is required
Address of the beneficiary's bank	+PmtInf ++CdtTrfTxInf +++CdtrAgt ++++FinInstnId +++++PstlAdr +++++AdrLine	M	70x	The field should be filled in if the BIC code is not given
Country of the beneficiary's bank	+PmtInf ++CdtTrfTxInf +++CdtrAgt ++++FinInstnId +++++PstlAdr +++++Ctry	M	2a!	
Name of the beneficiary	+PmtInf ++CdtTrfTxInf +++Cdtr ++++Nm	M	70x	
Address of the beneficiary	+PmtInf ++CdtTrfTxInf +++Cdtr +++PstlAdr ++++AdrLine	O	70x	Beneficiary address (removed)
Address of the beneficiary - Street	+PmtInf ++CdtTrfTxInf +++Cdtr +++PstlAdr ++++StrtNm	M If address is required	min 3x – max 24x	The previous field for the beneficiary's address (/AdrLine) is replaced with fields containing the address components described in the following lines. If an address is required for a given foreign transfer, the file must contain fields with its

				components, in accordance with their maturity and maintaining the minimum and maximum lengths specified below. If an address is not required for a given foreign transfer and the file contains a completed address field in the old structure, the address will be ignored. If the file contains an address in the new structure, it must meet the requirements and field length conditions. Address required for transfers: • Where the beneficiary bank country or beneficiary country is Canada, Switzerland, China, UK, Ukraine • transfers in EUR currency and with EXPRESS execution mode (HIGH+ SDVA parameters) • transfers in CNY currency (Yuan)
Address of the beneficiary - Building number	+PmtInf ++CdtTrfTxInf +++Cdtr +++PstlAdr ++++BldgNb	M If address is required	min 1x – max 3x	Building number
Address of the beneficiary - Flat number	+PmtInf ++CdtTrfTxInf +++Cdtr +++PstlAdr ++++FlatNb	O	min 0x – max 3x	Flat number
Address of the beneficiary - Postal code	+PmtInf ++CdtTrfTxInf +++Cdtr +++PstlAdr ++++PstCd	M If address is required	min 2x – max 6x	Postal code
Address of the beneficiary - Town name	+PmtInf ++CdtTrfTxInf +++Cdtr +++PstlAdr ++++TownNm	M If address is required	min 3x – max 19x	Town/City name
Address of the beneficiary - Code/Short name/Name of	+PmtInf ++CdtTrfTxInf +++Cdtr +++PstlAdr ++++CtrySubDvsn	M If address is required	min 2x – max 9x	Code/Short name/Name of the Region/District/Province

the Region/District/Province				
Country	+PmtInf ++CdtTrfTxInf +++Cdtr ++++PstlAdr +++++Ctry	O	2a!	Country code
Beneficiary's bank account number	+PmtInf ++CdtTrfTxInf +++CdtrAcct ++++Id +++++IBAN or +PmtInf ++CdtTrfTxInf +++CdtrAcct ++++Id +++++Othr +++++Id	M	2!a26!n or 26!n	In IBAN or NRB format
Order type	+PmtInf ++CdtTrfTxInf +++Purp ++++Prtry	M	4a!	Constant value "FRGN"
Title of the order	+PmtInf ++CdtTrfTxInf +++RmtInf ++++Ustrd	M	140x	

Example:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.03">
  <soapenv:Header/>
  <soapenv:Body>
    <CstmrCdtTrfInitn>
      <GrpHdr>
        <MsgId>Message ID</MsgId>
        <CreDtTm>2016-12-01T11:00:00</CreDtTm>
        <NbOfTxs> Number of transaction in message </NbOfTxs>
        <InitgPty>
          <Nm>Orderer name (np. użytkownik FK)</Nm>
        </InitgPty>
      </GrpHdr>
      <PmtInf>
        <PmtInfId>Folder name</PmtInfId>

```

```

<PmtMtd>TRF</PmtMtd>
<ReqdExctnDt> Execution date </ReqdExctnDt>
<Dbtr>
  <Nm>Orderer name</Nm>
  <PstlAdr>
    <AdrLine>Address</AdrLine>
    <AdrLine>Address continuation</AdrLine>
  </PstlAdr>
</Dbtr>
<DbtrAcct>
  <Id>
    <Othr>
      <Id> Bank account number </Id>
    </Othr>
  </Id>
</DbtrAcct>
<DbtrAgt>
  <FinInstnId>
    <ClrSysMmbld>
      <ClrSysId>
        <Cd>PLKNR</Cd>
      </ClrSysId>
      <Mmbld>Bank ID</Mmbld>
    </ClrSysMmbld>
  </FinInstnId>
</DbtrAgt>
<ChrgsAcct>
  <Id>
    <IBAN> Bank account number </IBAN>
  </Id>
</ChrgsAcct>
<CdtTrfTxInf>
  <PmtId>
    <InstrId>GOD</InstrId>
    <!-- "Business category" for China and CNY-->
  <EndToEndId> Transaction ID </EndToEndId>
  </PmtId>
  <PmtTpInf>
    <InstrPrty>HIGH</InstrPrty>
    <SvcLvl>
      <Cd>SDVA</Cd>
    </SvcLvl>
  </PmtTpInf>
  <Amt>
    <InstdAmt Ccy="EUR"> Amount </InstdAmt>
  </Amt>

```

```

<ChrgBr>SHAR</ChrgBr>
<CdtrAgt>
  <FinInstnId>
    <BIC>Kod SWIFT Banku beneficjenta</BIC>
    <Nm>Nazwa Banku beneficjenta</Nm>
    <PstlAdr>
      <StrtNm>Ulica</StrtNm>
      <BldgNb>123</BldgNb>
      <FlatNb>456</FlatNb>
      <PstCd>80331</PstCd>
      <TownNm>Miasto</TownNm>
      <CtrySubDvsn>BY</CtrySubDvsn>
      <Ctry>DE</Ctry>
    </PstlAdr>
  </FinInstnId>
</CdtrAgt>
<Cdtr>
  <Nm>Name of beneficiary</Nm>
  <PstlAdr>
    <Ctry>Kod kraju</Ctry>
    <AdrLine>Address</AdrLine>
  </PstlAdr>
</Cdtr>
<CdtrAcct>
  <Id>
    <IBAN> Bank account number </IBAN>
  </Id>
</CdtrAcct>
<Purp>
  <Prtry>FRGN</Prtry>
</Purp>
<RmtInf>
  <Ustrd> Title of the order </Ustrd>
</RmtInf>
</CdtTrfTxInf>
  </PmtInf>
</CstmrCdtTrfInitn>
</soapenv:Body>
</soapenv:Envelope>

```

3.3.1.1.6. TAX Payment

Field name	XML element name	Status	Format	Comments
Message ID	+PmtInf ++CdtTrfTxInf +++PmtId	M	16x	

	++++EndToEndId			
Order type	+PmtInf ++CdtTrfTxInf +++PmtTpInf ++++CtgyPurp +++++Cd	M	4a!	Constant value "TAXS"
Order amount	+PmtInf ++CdtTrfTxInf +++Amt ++++InstdAmt	M	15d	
Order currency	+PmtInf ++CdtTrfTxInf +++Amt ++++InstdAmt@Ccy	M	3a!	
Beneficiary's bank ID	+PmtInf ++CdtTrfTxInf +++CdtrAgt ++++FinInstnId +++++ClrSysMmbld ++++++Mmbld	M	8x	
Name of TAX Office	+PmtInf ++CdtTrfTxInf +++Cdtr ++++Nm	M	70x	
TAX Office's bank account number	+PmtInf ++CdtTrfTxInf +++CdtrAcct ++++Id +++++Othr ++++++Id	M	26!n	In NRB format
Payer ID	+PmtInf ++CdtTrfTxInf +++Tax ++++Dbtr +++++RegnId	M	1!x34x	The first character is "Identifier type": N - NIP P - PESEL R - Regon 1 - ID 2 - Passport 3 - other ID. The next one is the selected ID
Period	+PmtInf ++CdtTrfTxInf +++Tax ++++Rcrd +++++Tp	M	35x	The first part of the field is two characters intended for "Year" in the RR format. The second part contains one character - "Type of period":

				M - month, P - semester, R - year, K - quarter, D - decade, J - day. The third part contains the "Period Number." When "Period Type" assumes a value: R, this field does not contain the period number, P, this field is required to be filled by two characters with the values 01 or 02, K, this field is required to be filled by two characters with the values 01, 02, 03 or 04, M, this field is required to be filled by two characters with the values 01-12, D, this field is required to be filled by the first two characters with the values 01, 02 or 03, the next two values in the range 01-12, J, this field is required to be filled by two characters with values in the range 01 - 31 denoting the day and the next two characters of values from the 01 -12 range denoting the month
Tax form symbol	+PmtInf ++CdtTrfTxInf +++Tax ++++Rcrd +++++FrmsCd	M	35x	

Additional information	+PmtInf ++CdtTrfTxInf +++Tax ++++Rcrd +++++AddtlInf	O	40x	
Order type	+PmtInf ++CdtTrfTxInf +++Purp ++++Prtry	M	4a!	Constant value "TAXS"

Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
```

```
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.03">
```

```
<soapenv:Header/>
```

```
<soapenv:Body>
```

```
<CstmrCdtTrfInitn>
```

```
<GrpHdr>
```

```
<MsgId>Message ID</MsgId>
```

```
<CreDtTm>2016-12-01T11:00:00</CreDtTm>
```

```
<NbOfTx> Number of transaction in message </NbOfTx>
```

```
<InitgPty>
```

```
<Nm>Orderer name (np. użytkownik FK)</Nm>
```

```
</InitgPty>
```

```
</GrpHdr>
```

```
<PmtInf>
```

```
<PmtInfId>Folder name</PmtInfId>
```

```
<PmtMtd>TRF</PmtMtd>
```

```
<ReqdExctnDt> Execution date </ReqdExctnDt>
```

```
<Dbtr>
```

```
<Nm>Orderer name</Nm>
```

```
<PstlAdr>
```

```
<AdrLine>Address</AdrLine>
```

```
<AdrLine>Address continuation</AdrLine>
```

```
</PstlAdr>
```

```
</Dbtr>
```

```
<DbtrAcct>
```

```
<Id>
```

```
<Othr>
```

```
<Id> Bank account number </Id>
```

```
</Othr>
```

```
</Id>
```

```
</DbtrAcct>
```

```
<DbtrAgt>
```

```
<FinInstnId>
```

```
<ClrSysMmbld>
```

<ClrSysId>
 <Cd>PLKNR</Cd>
</ClrSysId>
 <Mmbld>Bank ID</Mmbld>
</ClrSysMmbld>
</FinInstnId>
</DbtrAgt>
 <CdtTrfTxInf>
 <PmtId>
 <EndToEndId> Transaction ID </EndToEndId>
 </PmtId>
 <PmtTpInf>
 <CtgyPurp>
 <Cd>TAXS</Cd>
 </CtgyPurp>
 </PmtTpInf>
 <Amt>
 <InstdAmt Ccy="PLN"> Amount </InstdAmt>
 </Amt>
 <CdtrAgt>
 <FinInstnId>
 <ClrSysMmbld>
 <Mmbld> Bank ID </Mmbld>
 </ClrSysMmbld>
 </FinInstnId>
 </CdtrAgt>
 <Cdtr>
 <Nm>Name of beneficiary</Nm>
 </Cdtr>
 <CdtrAcct>
 <Id>
 <Othr>
 <Id> Bank account number </Id>
 </Othr>
 </Id>
 </CdtrAcct>
 <Purp>
 <Prtry>TAXS</Prtry>
 </Purp>
 <Tax>
 <Dbtr>
 <RegnId>Identyfikator płatnika</RegnId>
 </Dbtr>
 <Rcrd>
 <Tp>Okres</Tp>

```

    <FrmsCd>INNE</FrmsCd>
    <AddtlInf>inny organ podatkowy</AddtlInf>
  </Rcrd>
</Tax>
</CdtTrfTxInf>
</PmtInf>
</CstmrCdtTrfInitn>
</soapenv:Body>
</soapenv:Envelope>

```

3.3.1.1.7. Social payment

Field name	XML element name	Status	Format	Comments
Message ID	+PmtInf ++CdtTrfTxInf +++PmtId ++++EndToEndId	M	16x	
Order type	+PmtInf ++CdtTrfTxInf +++PmtTpInf ++++CtgyPurp +++++Cd	M	4a!	Constant value "SSBE"
Order amount	+PmtInf ++CdtTrfTxInf +++Amt ++++InstdAmt	M	15d	
Order currency	+PmtInf ++CdtTrfTxInf +++Amt ++++InstdAmt@Ccy	M	3a!	
Beneficiary's bank ID	+PmtInf ++CdtTrfTxInf +++CdtrAgt ++++FinInstnId +++++ClrSysMmbld ++++++Mmbld	M	8x	
Name of Social Security Office	+PmtInf ++CdtTrfTxInf +++Cdtr ++++Nm	M	70x	Constant name: Zakład Ubezpieczeń Społecznych
Social Security Office's bank account number	+PmtInf ++CdtTrfTxInf +++CdtrAcct ++++Id +++++Othr ++++++Id	M	26!n	In NRB format

Payer's TAX ID	+PmtInf ++CdtTrfTxInf +++Tax ++++Dbtr +++++TaxId	M	35x	
Additional identifier	+PmtInf ++CdtTrfTxInf +++Tax ++++Dbtr +++++RegnId	M	1!x34x	The first character is " supplementary identifier type " : P - PESEL R - Regon 1 - ID 2 - Passport The next one is the selected ID
Payment type	+PmtInf ++CdtTrfTxInf +++Tax ++++Rcrd +++++Tp	M	35x	S - premium for 1 month, M - premium longer than 1 month, U - installment plan, T - postponement of the deadline, D - additional payment, E - execution, A - additional payment - error of the payer. The next 6 digits specify the "Period" of which it applies RRRRMM payment (YYYY - year, MM - month) The last 2 digits signify the "Declaration number" within the period
Number of decision / contract / enforceable title	+PmtInf ++CdtTrfTxInf +++Tax ++++Rcrd +++++AddtlInf	M	40x	
Order type	+PmtInf ++CdtTrfTxInf +++Purp ++++Prtry	M	4a!	Constant value "SSBE"

Example:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.03">
  <soapenv:Header/>
  <soapenv:Body>

```

```
<CstmrCdtTrfInittn>
  <GrpHdr>
    <MsgId>Message ID</MsgId>
    <CreDtTm>2016-12-01T11:00:00</CreDtTm>
    <NbOfTxs> Number of transaction in message </NbOfTxs>
    <InitgPty>
      <Nm>Orderer name (np. użytkownik FK)</Nm>
    </InitgPty>
  </GrpHdr>
  <PmtInf>
    <PmtInflId>Folder name</PmtInflId>
    <PmtMtd>TRF</PmtMtd>
    <ReqdExctnDt> Execution date </ReqdExctnDt>
    <Dbtr>
      <Nm>Orderer name</Nm>
      <PstlAdr>
        <AdrLine>Address</AdrLine>
        <AdrLine>Address continuation</AdrLine>
      </PstlAdr>
    </Dbtr>
    <DbtrAcct>
      <Id>
        <Othr>
          <Id> Bank account number </Id>
        </Othr>
      </Id>
    </DbtrAcct>
    <DbtrAgt>
      <FinInstnId>
        <ClrSysMmbld>
          <ClrSysId>
            <Cd>PLKNR</Cd>
          </ClrSysId>
          <Mmbld>Bank ID</Mmbld>
        </ClrSysMmbld>
      </FinInstnId>
    </DbtrAgt>
    <CdtTrfTxInf>
      <PmtId>
        <EndToEndId> Transaction ID </EndToEndId>
      </PmtId>
      <PmtTpInf>
        <CtgyPurp>
          <Cd>SSBE</Cd>
        </CtgyPurp>
```

```

</PmtTplnf>
<Amt>
  <InstdAmt Ccy="PLN"> Amount </InstdAmt>
</Amt>
<CdtrAgt>
  <FinInstnId>
    <ClrSysMmbld>
      <Mmbld> Bank ID </Mmbld>
    </ClrSysMmbld>
  </FinInstnId>
</CdtrAgt>
<Cdtr>
  <Nm>Zakład Ubezpieczeń Społecznych</Nm>
</Cdtr>
<CdtrAcct>
  <Id>
    <Othr>
      <Id> Bank account number </Id>
    </Othr>
  </Id>
</CdtrAcct>
<Purp>
  <Prtry>SSBE</Prtry>
</Purp>
<Tax>
  <Dbtr>
    <TaxId>NIP Płatnika</TaxId>
    <RegId>Identyfikator płatnika</RegId>
  </Dbtr>
  <Rcrd>
    <Tp>Typ płatności</Tp>
    <AddtlInf>Informacja dodatkowe</AddtlInf>
  </Rcrd>
</Tax>
</CdtTrfTxInf>
</PmtInf>
</CstmrCdtTrfInitn>
</soapenv:Body>
</soapenv:Envelope>

```

3.3.1.1.8. Postal order

Field name	XML element name	Status	Format	Comments
Reference	+PmtInf ++CdtTrfTxInf +++PmtId	0	105x	

	++++InstrId			
Message ID	+PmtInf ++CdtTrfTxInf +++PmtId ++++EndToEndId	M	16x	
Order amount	+PmtInf ++CdtTrfTxInf +++Amt ++++InstdAmt	M	15d	
Order currency	+PmtInf ++CdtTrfTxInf +++Amt ++++InstdAmt@Ccy	M	3a!	
Beneficiary's bank ID	+PmtInf ++CdtTrfTxInf +++CdtrAgt ++++FinInstnId +++++ClrSysMmbld ++++++Mmbld	M	8x	
Name of the beneficiary	+PmtInf ++CdtTrfTxInf +++Cdtr ++++Nm	M	70x	
Postal code of the beneficiary	+PmtInf ++CdtTrfTxInf +++PstlAdr ++++TwnNm	M	6x	
City of the beneficiary	+PmtInf ++CdtTrfTxInf +++PstlAdr ++++Ctry	M	25x	
Country	+PmtInf ++CdtTrfTxInf +++Cdtr ++++PstlAdr +++++Ctry	O	2a!	
Address of the beneficiary	+PmtInf ++CdtTrfTxInf +++Cdtr +++PstlAdr ++++AdrLine +++++AdrLine2 +++++AdrLine3	M	35x	AdrLine - street name AdrLine2 - house number AdrLine3 - apartment number

Abbreviation of the type of postal order	+PmtInf ++CdtTrfTxInf +++InstrForDbtrAgt	O	2a	Types of postal orders: 1. standard - abbreviation Z 2. standard with confirmation of receipt - abbreviation ZP, 3. Poste Restante - abbreviation R, 4. Poste Restante with confirmation of receipt - abbreviation RP
Order type	+PmtInf ++CdtTrfTxInf +++Purp ++++Prtry	M	4a!	Constant value "POST"
Title of the order	+PmtInf ++CdtTrfTxInf +++RmtInf ++++Ustrd	M	140x	

Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
```

```
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.03">
```

```
<soapenv:Header/>
```

```
<soapenv:Body>
```

```
<CstmrCdtTrfInitn>
```

```
<GrpHdr>
```

```
<MsgId>Message ID</MsgId>
```

```
<CreDtTm>2016-12-01T11:00:00</CreDtTm>
```

```
<NbOfTx> Number of transaction in message </NbOfTx>
```

```
<InitgPty>
```

```
<Nm>Orderer name (np. użytkownik FK)</Nm>
```

```
</InitgPty>
```

```
</GrpHdr>
```

```
<PmtInf>
```

```
<PmtInfId>Folder name</PmtInfId>
```

```
<PmtMtd>TRF</PmtMtd>
```

```
<ReqdExctnDt> Execution date </ReqdExctnDt>
```

```
<Dbtr>
```

```
<Nm>Orderer name</Nm>
```

```
<PstlAdr>
```

```
<Ctry>Kod kraju</Ctry>
```

```
<AdrLine>Address</AdrLine>
```

```
<AdrLine>Address continuation</AdrLine>
```

```
</PstlAdr>
```

```
</Dbtr>
```

```
<DbtrAcct>
```

```
<Id>
  <IBAN> Bank account number </IBAN>
</Id>
</DbtrAcct>
<DbtrAgt>
  <FinInstnId>
    <ClrSysMmbld>
      <ClrSysId>
        <Cd>PLKNR</Cd>
      </ClrSysId>
      <Mmbld> Bank ID </Mmbld>
    </ClrSysMmbld>
  </FinInstnId>
</DbtrAgt>
  <CdtTrfTxInf>
    <PmtId>
      <InstrId>Reference</InstrId>
      <EndToEndId>TR010</EndToEndId>
    </PmtId>
    <Amt>
      <InstdAmt Ccy="PLN"> Amount </InstdAmt>
    </Amt>
    <CdtrAgt>
      <FinInstnId>
        <ClrSysMmbld>
          <Mmbld>Bank ID</Mmbld>
        </ClrSysMmbld>
      </FinInstnId>
    </CdtrAgt>
    <Cdtr>
      <Nm>Kowalski Jan</Nm>
      <PstlAdr>
        <PstCd>Kod pocztowy</PstCd>
        <TwnNm>Miasto</TwnNm>
        <Ctry>PL</Ctry>
        <AdrLine>Address</AdrLine>
        <AdrLine2>Address</AdrLine2>
        <AdrLine3>Address</AdrLine3>
      </PstlAdr>
    </Cdtr>
    <InstrForDbtrAgt>Skrót rodzaju przekazu</InstrForDbtrAgt>
    <Purp>
      <Prtry>POST</Prtry>
    </Purp>
    <RmtInf>
```

```

    <Ustrd> Title of the order </Ustrd>
  </RmtInf>
</CdtTrfTxInf>
</PmtInf>
</CstmrCdtTrfInitn>
</soapenv:Body>
</soapenv:Envelope>

```

3.3.1.1.9. Prepaid - transfer from prepaid card account

Field name	XML element name	Status	Format	Comments
Message ID	+PmtInf ++CdtTrfTxInf +++PmtId ++++EndToEndId	M	16x	
Order amount	+PmtInf ++CdtTrfTxInf +++Amt ++++InstdAmt	M	15d	
Order currency	+PmtInf ++CdtTrfTxInf +++Amt ++++InstdAmt@Ccy	M	3a!	
Beneficiary's account number	+PmtInf ++CdtTrfTxInf +++CdtrAcct ++++Id +++++IBAN lub +PmtInf ++CdtTrfTxInf +++CdtrAcct ++++Id +++++Othr +++++Id	M	2!a26!n/26!n	In IBAN/NRB format
Order type	+PmtInf ++CdtTrfTxInf +++Purp ++++Prtry	M	4a!	Constant value "PREP"
Title of the order	+PmtInf ++CdtTrfTxInf +++RmtInf ++++Ustrd	M	140x	

Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.03">
  <soapenv:Header/>
  <soapenv:Body>
    <CstmrCdtTrfInitn>
      <GrpHdr>
        <MsgId>Identyfikator wiadomości</MsgId>
        <CreDtTm>2016-12-01T11:00:00</CreDtTm>
        <NbOfTxs> Number of transaction in message </NbOfTxs>
        <InitgPty>
          <Nm>Orderer name (np. użytkownik FK)</Nm>
        </InitgPty>
      </GrpHdr>
      <PmtInf>
        <PmtInfId>Folder name</PmtInfId>
        <PmtMtd>TRF</PmtMtd>
        <ReqdExctnDt> Execution date </ReqdExctnDt>
        <Dbtr>
          <Nm>przelew z kart PP</Nm>
        </Dbtr>
        <DbtrAcct>
          <Id>
            <Othr>
              <Id> Bank account number </Id>    <!-- numer rach karty przedpłaconej 5427 XXXX XXXX 2909 -->
            </Othr>
          </Id>
        </DbtrAcct>
        <DbtrAgt>
          <FinInstnId>
            <ClrSysMmbld>
              <ClrSysId>
                <Cd>PLKNR</Cd>
              </ClrSysId>
              <Mmbld> Bank ID </Mmbld>
            </ClrSysMmbld>
          </FinInstnId>
        </DbtrAgt>
        <CdtTrfTxInf>
          <PmtId>
            <EndToEndId> Transaction ID </EndToEndId>
          </PmtId>
          <Amt>
            <InstdAmt Ccy="PLN"> Amount </InstdAmt>
          </Amt>
        </CdtTrfTxInf>
      </PmtInf>
    </CstmrCdtTrfInitn>
  </soapenv:Body>
</soapenv:Envelope>
```

```

<CdtrAgt>
  <FinInstnId>
    <ClrSysMmbld>
      <Mmbld> Bank ID </Mmbld>
    </ClrSysMmbld>
  </FinInstnId>
</CdtrAgt>
<CdtrAcct>
  <Id>
    <Othr>
      <Id> Bank account number </Id>
    </Othr>
  </Id>
</CdtrAcct>
<Purp>
  <Prtry>PREP</Prtry>
</Purp>
<RmtInf>
  <Ustrd> Title of the order </Ustrd>
</RmtInf>
</CdtTrfTxInf>
</PmtInf>
</CstmrCdtTrfInitn>
</soapenv:Body>
</soapenv:Envelope>

```

* VAT invoice number (including correcting invoice) or - for "accelerated" returns made by the tax office - the period for which the refund is requested or - in the case of intra-Community acquisition of goods - the number of the document related to the payment issued by the payer or - in the case of transfer to the VAT account of the VAT account holder kept in the same bank - the words "Przekazanie własne"

** the number by which the supplier of goods or service provider is identified for the purpose of tax or - for "accelerated" returns made by the tax office - the number by which the taxpayer is identified for the purposes of tax or - in the case of intra-Community acquisition of goods - by number whose payer is identified for the purposes of tax or - for payments regarding corrections of VAT invoices - the number by which the recipient or customer is identified for the purposes of tax or - in the case of transferring funds to the VAT account of the VAT account holder in the same bank - by which the holder is identified for the purposes of tax

3.3.1.2. CreditTransferResponse

Message received as an automatic response to sent CreditTransfer messages. Allows to verify how many payments have been sent. In addition, the message shows the number of erroneous orders or informs that the entire CreditTransfer message has been rejected.

Message structure

Field name	XML element name	Status	Format	Comments
------------	------------------	--------	--------	----------

Reference number	+CstmrPmtStsRpt ++GrpHdr +++MsgId	M	32x	The number assigned by the Bank
Time stamp for sending the package	+CstmrPmtStsRpt ++GrpHdr +++CreDtTm	M	RRRR-MM-DDThh:mm:ss	
CreditTransfer request ID	+CstmrPmtStsRpt ++OrgnlGrpInfAndSts +++OrgnlMsgNmId	M	36x	The number assigned by the Bank
Number of payments	+CstmrPmtStsRpt ++OrgnlGrpInfAndSts +++OrgnlNbOfTxS	M	5n	
Payment ID of the declined payment	+CstmrPmtStsRpt ++TxInfAndSts +++OrgnlEndToEndId	C	16x	The reference number given by the Customer
Status	+CstmrPmtStsRpt ++TxInfAndSts +++TxSts	C	4a	Constant value of "RJCT", the status appears only when errors occur
Error description	+CstmrPmtStsRpt ++TxInfAndSts +++StsRsnInf ++++AddtlStsRsnInf	C	no limit	A text description of the error of the rejected order may occur many times for a single order

Example:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <CstmrPmtStsRpt xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.02">
      <GrpHdr>
        <MsgId>Message ID</MsgId>
        <CreDtTm>2017-11-02T11:22:18.1057201+01:00</CreDtTm>
      </GrpHdr>
      <OrgnlGrpInfAndSts>
        <OrgnlMsgNmId>CreditTransfer request ID</OrgnlMsgNmId>
        <OrgnlNbOfTxS>Number of transaction in message</OrgnlNbOfTxS>
      </OrgnlGrpInfAndSts>
    </CstmrPmtStsRpt>
  </s:Body>
</s:Envelope>

```

Example with error:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

```

```

<s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <CstmrPmtStsRpt xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.02">
    <GrpHdr>
      <MsgId>Message ID</MsgId>
      <CreDtTm>2017-11-02T10:55:04.1103052+01:00</CreDtTm>
    </GrpHdr>
    <OrgnlGrpInfAndSts>
      <OrgnlNbOfTx>Number of transaction in message</OrgnlNbOfTx>
    </OrgnlGrpInfAndSts>
    <TxInfAndSts>
      <OrgnlEndToEndId>ID płatności odrzuconej</OrgnlEndToEndId>
      <TxSts>RJCT</TxSts>
      <StsRsnInf>
        <AddtlStsRsnInf>Przykład odpowiedzi: Zlecenie ma datę wykonania wcześniejszą niż dzień dzisiejszy
lub dalszą niż 11 miesięcy.</AddtlStsRsnInf>
      </StsRsnInf>
      <StsRsnInf>
        <AddtlStsRsnInf>Przykład odpowiedzi: Kwota zlecenia powinna być wyższa od: '0.00'</AddtlStsRsnInf>
      </StsRsnInf>
    </TxInfAndSts>
  </CstmrPmtStsRpt>
</s:Body>
</s:Envelope>

```

3.3.2. GetPaymentStatusReport

3.3.2.1. GetPaymentStatusReportRequest

Inquiry sent to check the status of payments sent in the CreditTransfer message

Message structure

Field name	XML element name	Status	Format	Comments
Message ID	GetPayStsRpt +GrpHdr ++MsgId +++Id	M	16x	The number assigned by the Customer
The identifier of the original CreditTransfer query	GetPayStsRpt + OrgnlGrpInfAndSts ++OrgnlMsgId	M	36x	The number passed in response to CreditTransfer in the <OrgnlMsgNmId> field

Example:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:get="http://www.bankmillennium.pl/CorporateServices/GetPaymentStatusReport">
  <soapenv:Header/>

```

```

<soapenv:Body>
  <get:GetPayStsRpt>
    <get:GrpHdr>
      <get:MsgId>
        <get:Id>Message ID</get:Id>
      </get:MsgId>
    </get:GrpHdr>
    <get:OrgnlGrpInfAndSts>
      <get:OrgnlMsgId>Identyfikator oryginalnego zapytania CreditTransfer</get:OrgnlMsgId>
    </get:OrgnlGrpInfAndSts>
  </get:GetPayStsRpt>
</soapenv:Body>
</soapenv:Envelope>

```

3.3.2.2. GetPaymentStatusReportResponse

Message received as an automatic response to sent GetPaymentStatusReportRequest messages. Allows to verify the status of transactions sent.

Message structure

Field name	XML element name	Status	Format	Comments
Message ID	CstmrPmtStsRpt +GrpHdr ++MsgId	M	16x	The number assigned by the bank
Time stamp	CstmrPmtStsRpt +GrpHdr ++CreDtTm	M	RRRR-MM-DDThh:mm:ss+GMT+1	
The number of the original CreditTransfer message	CstmrPmtStsRpt +OrgnlGrpInfAndSts ++OrgnlMsgNmId	M	36x	The number passed in response to CreditTransfer in the <OrgnlMsgNmId> field
The number of original transactions	CstmrPmtStsRpt +OrgnlGrpInfAndSts ++OrgnlNbOfTxS	M	n	The number of transactions sent in the original CreditTransfer
Message ID	CstmrPmtStsRpt +TxInfAndSts ++OrgnlEndToEndId	M	16x	The number assigned by the Customer
Transaction status	CstmrPmtStsRpt +TxInfAndSts ++TxSts	M	4a	Details in Codes and status description table
Description of transaction status	CstmrPmtStsRpt +TxInfAndSts ++StsRsnInf	M	140x	Details in Codes and status description table

Codes and status descriptions table

TxSts	AddtlStsRsnInf
ACSP	AcceptedForExecution
ACSP	Authorized
ACSP	Executed
ACSP	Processing
ACSP	Posted
PDNG	OrderOk
PDNG	DraftValid
PDNG	FutureDate
PDNG	OnHold
PDNG	Importing
PDNG	Initialized
PDNG	Liquidity
PDNG	Locked
PDNG	PartiallyAuthorized
PDNG	RequiresBankAuthorization
PDNG	Recovery
PDNG	Sending
PDNG	Unknown
PDNG	Validating
PDNG	WaitingAuthorization
RJCT	OrderStatus.Aborted
RJCT	OrderNotOk
RJCT	AuthorizationCanceled
RJCT	Deleted
RJCT	DraftInvalid
RJCT	Expired
RJCT	Failed
RJCT	RecycleBin
RJCT	Returned
RJCT	Refused
RJCT	Rejected

Example:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <CstmrPmtStsRpt xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.02">
      <GrpHdr>
        <MsgId>Message ID</MsgId>
        <CreDtTm>2017-11-17T16:36:17.7757472+01:00</CreDtTm>
      </GrpHdr>
      <OrgnlGrplnfAndSts>

```

```

    <OrgnlMsgNmId>Numer oryginalnej wiadomości</OrgnlMsgNmId>
    <OrgnlNbOfTx>Number of transaction in message</OrgnlNbOfTx>
  </OrgnlGrpInfAndSts>
  <TxInfAndSts>
    <OrgnlEndToEndId>Transaction ID</OrgnlEndToEndId>
    <TxSts>Transaction status description</TxSts>
    <StsRsnInf>
      <AddtlStsRsnInf>Trensaction status description</AddtlStsRsnInf>
    </StsRsnInf>
  </TxInfAndSts>
  <TxInfAndSts>
    <OrgnlEndToEndId>Transaction ID</OrgnlEndToEndId>
    <TxSts> Transaction status</TxSts>
    <StsRsnInf>
      <AddtlStsRsnInf>Transaction status description</AddtlStsRsnInf>
    </StsRsnInf>
  </TxInfAndSts>
</CstmrPmtStsRpt>
</s:Body>
</s:Envelope>

```

3.4. Reports

3.4.1. GetReportList

The GetReportList method supports the exchange of information between the Customer and the Bank regarding the list of selected types of periodical reports.

3.4.1.1. GetReportListRequest

Sent by the Customer to the Bank. It contains parameters regarding the identification of a given report, the account concerned and its date range.

Message structure:

Field name	XML element name	Status	Format	Comments
Message ID	GetRptLst +MsgId ++Id	O	16x	Unique for a given report. Can not contain spaces.
Report type	GetRptLst +RptLstQryDef ++RptLstCrit +++NewCrit ++++SchCrit +++++RptType	M	String	Report type. Possible values are: AIPP - AIPP accounts report PrepaidCard - List of prepaid cards

				PrepaidDaily - Daily list of transactions for pre-paid cards PrepaidMonthly - Monthly statement of operations from prepaid cards
Additional indicator for the report	GetRptLst +RptLstQryDef ++RptLstCrit +++NewCrit ++++SchCrit +++++RptTKind	M	2!a26!n	Account number in the IBAN format in the case of the AIPP report The field may remain empty for reports for pre-paid cards (PrepaidCard, PrepaidDaily, PrepaidMonthly)
Date from	GetRptLst +RptLstQryDef ++RptLstCrit +++NewCrit ++++SchCrit +++++RptType +++++RptValDt +++++DtSch +++++FrDt	M	YYYY-MM-DD	
Date to	GetRptLst +RptLstQryDef ++RptLstCrit +++NewCrit ++++SchCrit +++++RptType +++++RptValDt +++++DtSch +++++ToDt	M	YYYY-MM-DD	

Example:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:tem="http://www.bankmillennium.pl/CorporateServices"
  xmlns:urn="http://www.bankmillennium.pl/CorporateServices/Reports/GetReportListRequest">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:GetReportList>
      <tem:request>
        <urn:GetRptLst>
          <urn:MsgId>

```

```

        <urn:Id>Message ID</urn:Id>
    </urn:MsgId>
    <urn:RptLstQryDef>
        <urn:RptLstCrit>
            <urn:NewCrit>
                <urn:SchCrit>
                    <!--Report type- AIPP, PrepaidCard,
PrepaidDaily, PrepaidMonthly-->
                    <urn:RptType>Report type</urn:RptType>
                    <!--Additional data for the report - for AIPP
report: IBAN, for Prepaid reports may be empty-->
                    <urn:RptKind>Bank account number</urn:RptKind>
                    <urn:RptValDt>
                        <urn:DtSch>
                            <urn:FrDt>Date from</urn:FrDt>
                            <urn:ToDt>Date to</urn:ToDt>
                        </urn:DtSch>
                    </urn:RptValDt>
                </urn:SchCrit>
            </urn:NewCrit>
        </urn:RptLstCrit>
    </urn:RptLstQryDef>
</urn:GetRptLst>
</tem:request>
</tem:GetReportList>
</soapenv:Body>
</soapenv:Envelope>

```

3.4.1.2. GetReportListResponse

Sent by the Bank to the Customer. It contains a list of reports of a given type generated by the Bank, for a given account in a given period

Message structure:

Field name	XML element name	Status	Format	Comments
Message ID	RptLst +MsgId ++Id	O	String	Unique for a given report. Can not contain spaces.
Report Name	RptLst +LstNtry ++RptNm	M	String	
Date of the generation of the report	RptLst +LstNtry ++CreDt	M	YYYY-MM-DD	

Chosen type of the report	RptLst +LstNtry ++RptDtls +++RptType	M	String	
File size	RptLst +LstNtry ++RptDtls +++RptSize	M	String	Field visible only for the report type: PrepaidCard, Prepaid Daily, Prepaid Monthly
Report ID number	RptLst +LstNtry ++RptDtls +++RptId	M	String	

Example:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <GetReportListResponse xmlns="http://www.bankmillennium.pl/CorporateServices">
      <GetReportListResult>
        <RptLst xmlns="http://www.bankmillennium.pl/CorporateServices/Reports/GetReportListResponse">
          <MsgId>
            <Id>Message ID</Id>
          </MsgId>
          <LstNtry>
            <RptNm>Report name</RptNm>
            <CreDt>Date of the report</CreDt>
            <RptDtls>
              <RptType>Report type</RptType>
              <RptId>Report ID</RptId>
            </RptDtls>
          </LstNtry>
        </RptLst>
      </GetReportListResult>
    </GetReportListResponse>
  </s:Body>
</s:Envelope>

```

3.4.2. GetReport

The GetReport method supports the exchange of information between the Customer and the Bank regarding a specific report selected by the Client

3.4.2.1. GetReportRequest

Sent by the Customer to the Bank. It contains parameters that allow you to identify the report

Message structure:

Field name	XML element name	Status	Format	Comments
Message ID	GetRpt +MsgId ++Id	O	16x	Unique for a given report. Can not contain spaces.
Report ID number	GetRpt +RptQryDef ++RptCrit +++NewCrit ++++SchCrit +++++RptId ++++++EQ	M	String	Report ID from the GetReportList

Example:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:tem="http://www.bankmillennium.pl/CorporateServices"
  xmlns:urn="http://www.bankmillennium.pl/CorporateServices/Reports/GetReportRequest">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:GetReport>
      <tem:request>
        <urn:GetRpt>
          <urn:MsgId>
            <urn:Id>Message ID</urn:Id>
          </urn:MsgId>
          <urn:RptQryDef>
            <urn:RptCrit>
              <urn:NewCrit>
                <urn:SchCrit>
                  <urn:RptId>
                    <!-- Report ID from the GetReportList response -->
                    <urn:EQ> Report ID from the GetReportList response </urn:EQ>
                  </urn:RptId>
                </urn:SchCrit>
              </urn:NewCrit>
            </urn:RptCrit>
          </urn:RptQryDef>
        </urn:GetRpt>
      </tem:request>
    </tem:GetReport>
  </soapenv:Body>
</soapenv:Envelope>

```

3.4.2.2. GetReportResponse

Sent by the Bank to the Customer. It will contain reports of a given type in the form of Base64Binary files in the format given by the Bank

Message structure:

Field name	XML element name	Status	Format	Comments
Message ID	Rpt +MsgId ++Id	O	String	Unique for a given report. Can not contain spaces.
Report name	Rpt +RptDtls ++RptNm	M	String	
File size	Rpt +RptDtl ++RptSize	M	String	
File in Base64Binary format	Rpt +RptDtls ++RptFile	M	String	

Example:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <GetReportResponse xmlns="http://www.bankmillennium.pl/CorporateServices">
      <GetReportResult>
        <Rpt xmlns="http://www.bankmillennium.pl/CorporateServices/Reports/GetReportResponse">
          <MsgId>
            <Id>Message ID</Id>
          </MsgId>
          <RptDtls>
            <RptNm>Report name</RptNm>
            <RptSize>Report file size</RptSize>
            <RptFile>File in Base64Binary format</RptFile>
          </RptDtls>
        </Rpt>
      </GetReportResult>
    </GetReportResponse>
  </s:Body>
</s:Envelope>
```

3.4.3. GetRecentReport

The GetRecentReport method supports the exchange of information between the Customer and the Bank in the Client's reports.

It differs from GetReport in that reports are always downloaded since the last call of this service for a given report, which means that the client does not complete the date range.

The moment from which the method starts to store data is the date of establishment of the Virtual User.

3.4.3.1. GetRecentReportRequest

Sent by the Customer to the Bank. Contains the type of report that the client expects

Message structure:

Field name	XML element name	Status	Format	Comments
Message ID	GetRecentRpt +MsgId ++Id	O	16x	Unique for a given report. Can not contain spaces.
Report type	GetRecentRpt +RctRptQryDef ++RctRptCrit +++NewCrit ++++SchCrit +++++RptType	M	String	Report type. Possible values are: AIPP - AIPP accounts report PrepaidCard - List of prepaid cards PrepaidDaily - Daily list of transactions for pre-paid cards PrepaidMonthly - Monthly statement of operations from prepaid cards
Additional indicator for the report	GetRecentRpt +RctRptQryDef ++RctRptCrit +++NewCrit ++++SchCrit +++++RptKind	M	2!a26!n	Account number in the IBAN format in the case of the AIPP report The field may remain empty for reports for pre-paid cards (PrepaidCard, PrepaidDaily, PrepaidMonthly)

Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:tem="http://www.bankmillennium.pl/CorporateServices" xmlns:get="GetRecentDocumentRequest">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:GetRecentReport>
      <tem:request>
```

```

<get:GetRecentRpt>
  <get:MsgId>
    <get:Id>Message ID</get:Id>
  </get:MsgId>
  <get:RctRptQryDef>
    <get:RctRptCrit>
      <get:NewCrit>
        <get:SchCrit>
          <get:RptType>Report type</get:RptType>
          <get:RptKind> Additional indicator for the report </get:RptKind>
        </get:SchCrit>
      </get:NewCrit>
    </get:RctRptCrit>
  </get:RctRptQryDef>
</get:GetRecentRpt>
</tem:request>
</tem:GetRecentReport>
</soapenv:Body>
</soapenv:Envelope>

```

3.4.3.2. GetRecentReportResponse

Sent by the Bank to the Customer. The structure does not differ from GetReport. Includes incremental reports since the last time the service was called.

Message structure:

Field name	XML element name	Status	Format	Comments
Message ID	Rpt +MsgId ++Id	O	String	Unique for a given report. Can not contain spaces.
Report name	Rpt +RptDtls ++DocumentDetails +++RptNm	M	String	
File size	Rpt +RptDtl ++DocumentDetails +++RptSize	M	String	
File in Base64Binary format	Rpt +RptDtls ++DocumentDetails +++RptFile	M	String	

Example:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">

```

```

<GetRecentReportResponse xmlns="http://www.bankmillennium.pl/CorporateServices">
  <GetRecentReportResult>
    <Rpt xmlns="GetRecentDocumentResponse">
      <MsgId>
        <Id>Message ID</Id>
      </MsgId>
      <RptDtls>
        <DocumentDetails>
          <RptNm>Report name</RptNm>
          <RptSize>Report file size</RptSize>
          <RptFile>File in Base64Binary format</RptFile>
        </DocumentDetails>
      </RptDtls>
    </Rpt>
  </GetRecentReportResult>
</GetRecentReportResponse>
</s:Body>
</s:Envelope>

```

3.5. Factoring

3.5.1. InvoiceBatch

The InvoiceBatch method supports the exchange of information between the Customer and the Bank regarding the list of invoices with receivables.

3.5.1.1. InvoiceBatchRequest

sent by the Customer to the Bank. It contains a file with invoices in Base64binary format compatible with the template defined by the Bank (it contains, among others, a checksum).

Message structure:

Field name	XML element name	Status	Format	Comments
Frame agreement number	InvoiceBatch + request ++AgreementNumber	M	String	
File with invoices	InvoiceBatch +request ++InvoiceFile	M	String - base64	

Example:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:cor="http://www.bankmillennium.pl/CorporateServices">

```

```

<soapenv:Header/>
<soapenv:Body>
  <cor:InvoiceBatch>
    <!--Optional:-->
    <cor:request>
      <!--Optional:-->
      <cor:AgreementNumber>Frame agreement number</cor:AgreementNumber>
      <!--Optional:-->
      <cor:InvoiceFile>File in base64 format</cor:InvoiceFile>
    </cor:request>
  </cor:InvoiceBatch>
</soapenv:Body>
</soapenv:Envelope>

```

3.5.1.2. InvoiceBatchResponse

Sent by the Bank to the Customer. It contains the package number assigned by the system and its status

Message structure:

Field name	XML element name	Status	Format	Comments
Package number	InvoiceBatchResult +PackageNumber	M	String - guid	Unikalny numer dla każdej paczki nadany przez Bank
Status	InvoiceBatchResult +Message	M	String	Informacja zwrotna na temat wysłanej paczki z przelewami

Example:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <InvoiceBatchResponse xmlns="http://www.bankmillennium.pl/CorporateServices">
      <InvoiceBatchResult>
        <PackageNumber>Package number - guid</PackageNumber>
        <Message> The package has been accepted for processing. </Message>
      </InvoiceBatchResult>
    </InvoiceBatchResponse>
  </s:Body>
</s:Envelope>

```

3.5.2. GetInvoiceBatchStatusReport

The GetInvoiceBatchStatusReport method supports the exchange of information between the Customer and the Bank in terms of statuses for the list of invoices with debts.

3.5.2.1. GetInvoiceBatchStatusReportRequest

Sent by the Customer to the Bank. Includes individual package number assigned by the Bank.

Message structure:

Field name	XML element name	Status	Format	Comments
Package number given by Bank	GetInvoiceBatchStatusReport +request ++PackageNumber	M	String-guid	Numer paczki nadany przez Bank w odpowiedzi do InvoiceBatch w polu: <i>PackageNumber</i>

Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:cor="http://www.bankmillennium.pl/CorporateServices">
  <soapenv:Header/>
  <soapenv:Body>
    <cor:GetInvoiceBatchStatusReport>
      <!--Optional:-->
      <cor:request>
        <cor:PackageNumber>Package number</cor:PackageNumber>
      </cor:request>
    </cor:GetInvoiceBatchStatusReport>
  </soapenv:Body>
</soapenv:Envelope>
```

3.5.2.2. GetInvoiceBatchStatusReportResponse

Sent by the Bank to the Customer. Contains file in base64 format with statuses for given invoices.

Message structure:

Field name	XML element name	Status	Format	Comments
Package number	GetInvoiceBatchStatusReportResult +PackageNumber	M	String - guid	Unikalny numer dla każdej paczki
Status	GetInvoiceBatchStatusReportResult +PaymentReportStatusFile	M	String - base64	Plik zawierający Informację zwrotną na temat wysłanej paczki z przelewami w formacie base64

Example:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

```

<GetInvoiceBatchStatusReportResponse xmlns="http://www.bankmillennium.pl/CorporateServices">
  <GetInvoiceBatchStatusReportResult>
    <PackageNumber>Package number</PackageNumber>
    <PaymentReportStatusFile>Report file in base64 format</PaymentReportStatusFile>
  </GetInvoiceBatchStatusReportResult>
</GetInvoiceBatchStatusReportResponse>
</s:Body>
</s:Envelope>

```

3.5.3. FactoringReports

The FactoringReports method supports the exchange of information between the Customer and the Bank in the scope of reports defined by the Bank.

3.5.3.1. FactoringReportsRequest

Sent by the Customer to the Bank. Contains the indication of the specific report to download.

Message structure:

Field name	XML element name	Status	Format	Comments
Report ID	FactoringReports +request ++ReportId	M	String	Numer udostępnionego dla Klienta raportu podawany będzie przez Bank
Franchise agreement number	FactoringReports +request ++AgreementNumber	M	String	
Report filter	FactoringReports +request ++FilterParametersFile	M		Filtr raportu, podany przez Bank w formacie Base64

Example:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:cor="http://www.bankmillennium.pl/CorporateServices">
  <soapenv:Header/>
  <soapenv:Body>
    <cor:FactoringReports>
      <!--Optional:-->
      <cor:request>
        <cor:ReportId>Report ID</cor:ReportId>
        <!--Optional:-->
        <cor:AgreementNumber>Agreement number</cor:AgreementNumber>
        <!--Optional:-->
        <cor:FilterParametersFile>Report filter in base64 format</cor:FilterParametersFile>
      </cor:request>
    </cor:FactoringReports>
  </soapenv:Body>
</soapenv:Envelope>

```

```

    </cor:request>
  </cor:FactoringReports>
</soapenv:Body>
</soapenv:Envelope>

```

3.5.3.2. FactoringReportsResponse

Sent by the Bank to the Customer. Contains report in base64 format

Message structure:

Field name	XML element name	Status	Format	Comments
Report status	FactoringReportsResult +Message	M	String	Status raportu
Report file	FactoringReportsResult +ReportFile	M	String-guid	Plik z raportem w formacie base64

Example:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <FactoringReportsResponse xmlns="http://www.bankmillennium.pl/CorporateServices">
      <FactoringReportsResult>
        <Message>Report status</Message>
        <ReportFile>Report file in base64 format</ReportFile>
      </FactoringReportsResult>
    </FactoringReportsResponse>
  </s:Body>
</s:Envelope>

```

3.6. Qualified signature

3.6.1. Functional description

CreditTransfer webservice method from /PaymentsService.svc endpoint supports payment authorization using one or multiple qualified electronic signatures. Signatures must be passed in SOAP message header named Signatures (marked yellow in the request example below). In this header one or more XAdES signatures can be passed.

Signature must be in XAdES-BES (Basic Electronic Signature) detached format. Notice that Reference element which contains digest of signed document must have URI attribute which is not empty (marked red in the request example below).

The document which is digitally signed is CreditTransfer SOAP request body (XML document with root CstmrCdtTrfInit marked blue in the request example below). The document must have only one default

namespace which value is "urn:iso:std:iso:20022:tech:xsd:pain.001.001.03" and it must not have any namespace prefix. Before signing document must be encoded using UTF-8 encoding.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:iso:std:iso:20022:tech:xsd:pain.001.001.03">
```

```
<soapenv:Header>
```

```
<Signatures>
```

```
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="ID-4d3ddf17-c74e-4ae4-bc2a-cb9c3fef9636">
```

```
<ds:SignedInfo Id="ID-11fb51fa-c8f8-4d99-98d1-41796f6b0d2f">
```

```
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
```

```
<ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
```

```
<ds:Reference Id="ID-eed3aa07-3add-4e94-9748-51e4e5ffe4c7" URI="UATRequest.xml">
```

```
<ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#sha256"/>
```

```
<ds:DigestValue>nj2tW8MeS5AOnq50BcwFjr2nermq7N3k/cx7tLWWasg=</ds:DigestValue>
```

```
</ds:Reference>
```

```
<ds:Reference Id="ID-686bf799-bf5b-4683-9ea6-c15da8a11b2b" URI="#ID-0d176481-c486-41f9-a269-f55dd250d32c" Type="http://uri.etsi.org/01903#SignedProperties">
```

```
<ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#sha256"/>
```

```
<ds:DigestValue>SlayNY9ikMgEtpO6e741stAv3FamStIFONEJaH0BL+Q=</ds:DigestValue>
```

```
</ds:Reference>
```

```
</ds:SignedInfo>
```

```
<ds:SignatureValue Id="ID-649ddc28-f499-4d03-b450-
```

```
66dfc42bc95f">aMtVMKC7GtVJxjHriSKJ3OofVNdNKnyl+/jLJz9jrr1/0J73D9ukPuyj5pSGkq6XPbW/uSOO/ESV9aSx8e
3BB6mFWyvr3D2XDXX4roDvihruw7rkcmqUcmshK3vgynira7HfMFF65922Rx2+lwyMFzMywG221023n0Pxz2EleylJmz
QUJEWJrO9PE2kabzMChdnCWAiwjtsHRyTGrTdp6fRmYNqnRNY8CAHe0LecStia/RJg/k7yKLgEFGyBj3EPRFKfGWS66
gpnFCdF6mxRVBXJT8hxhw2WEJAg/wn8xewy9C0sOYxNc2uU5DnW8V5NeCNm6z9/R2virV8M1VLzA==</ds:Signatur
eValue>
```

```
<ds:KeyInfo>
```

```
<ds:KeyValue>
```

```
<ds:RSAKeyValue>
```

```
<ds:Modulus>AJgPpEKF/1qbonKpDKoaH1SK6dvcb83TU4X5pM6ZtGVR0+BbsFLYxDo0vC+0iClkjG
Fdbt67oFJQGQf0xSHeWQW1rgldg5PThT4jWvN2Huzfnkan39qy1+yM1uHBFrPsuE4yzATEe9PZt9Zv1B7hG/QdwTQyG
maL0B32bwuh5sEJFf+10d/5YsOV+r4Pu7yEbPg1lGorkKNS7ul47p0MJWmLuSLgINA5702Im5GFf6rzyxrGpvnGJz31yK
H6RIkIIPhfDmK8agyhRIEeeOZrydlujQwly3XxQemfGgxyXYozh3igYgQqzuHG19tFablRR9d0W4jqO1qKKyarlCHFPE=<
/ ds:Modulus>
```

```
<ds:Exponent>AQAB</ds:Exponent>
```

```
</ds:RSAKeyValue>
```

```
</ds:KeyValue>
```

```
<ds:X509Data>
```

```
<ds:X509CertificateXXXXXXXXXX</ds:X509Certificate>
```

```
</ds:X509Data>
```

```
</ds:KeyInfo>
```

```
<ds:Object>
```

```

    <xades:QualifyingProperties xmlns:xades="http://uri.etsi.org/01903/v1.3.2#" Id="ID-7ab70ecd-8c35-4b67-9c12-2f72d4aec193" Target="#ID-4d3ddf17-c74e-4ae4-bc2a-cb9c3fef9636">
      <xades:SignedProperties Id="ID-0d176481-c486-41f9-a269-f55dd250d32c">
        <xades:SignedSignatureProperties>
          <xades:SigningTime>2019-05-23T12:44:15Z</xades:SigningTime>
          <xades:SigningCertificate>
            <xades:Cert>
              <xades:CertDigest>
                <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig#sha256"/>
                <ds:DigestValue>x/DAYjz9gEdxbi3r8rIWFY/Aw8Hl75ZbTa7hq+l2s8U=</ds:DigestValue>
              </xades:CertDigest>
            </xades:Cert>
            <xades:IssuerSerial>
              <ds:X509IssuerName>organizationIdentifier=VATPL-5260300517,CN=COPE SZAFIR -Kwalifikowany,O=Krajowa Izba Rozliczeniowa S.A.,C=PL</ds:X509IssuerName>
              <ds:X509SerialNumber>396555181370227369019555485616523982939339981406</ds:X509SerialNumber>
            </xades:IssuerSerial>
          </xades:Cert>
        </xades:SigningCertificate>
      </xades:SignedSignatureProperties>
      <xades:SignedDataObjectProperties>
        <xades:DataObjectFormat ObjectReference="#ID-eed3aa07-3add-4e94-9748-51e4e5ffe4c7">
          <xades:Description>Dokument w formacie xml [XML]</xades:Description>
          <xades:MimeType>application/octet-stream</xades:MimeType>
        </xades:DataObjectFormat>
        <xades:CommitmentTypeIndication>
          <xades:CommitmentTypeInd>
            <xades:Identifier>http://uri.etsi.org/01903/v1.2.2#ProofOfApproval</xades:Identifier>
          </xades:CommitmentTypeInd>
        </xades:CommitmentTypeIndication>
      </xades:SignedDataObjectProperties>
    </xades:SignedProperties>
  </xades:QualifyingProperties>
</ds:Object>
</ds:Signature>
</Signatures>
</soapenv:Header>
<soapenv:Body>
  <CstmrCdtTrfInittn xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.03">
    <GrpHdr>
      <MsgId>Credit transfer</MsgId>

```

```
<CreDtTm>2016-04-01T11:00:00</CreDtTm>
<NbOfTx>1</NbOfTx>
<InitgPty>
  <Nm>Orderer name</Nm>
</InitgPty>
</GrpHdr>
<PmtInf>
  <PmtInfId>Signature tests</PmtInfId>
  <PmtMtd>TRF</PmtMtd>
  <ReqdExctnDt>2019-05-23</ReqdExctnDt>
  <Dbtr>
    <Nm>Company Name</Nm>
    <PstlAdr>
      <AdrLine>Street 1</AdrLine>
      <AdrLine>00-100 City</AdrLine>
    </PstlAdr>
  </Dbtr>
  <DbtrAcct>
    <Id>
      <Othr>
        <Id>92116022020000000111379980</Id>
      </Othr>
    </Id>
  </DbtrAcct>
  <DbtrAgt>
    <FinInstnId>
      <ClrSysMmbld>
        <ClrSysId>
          <Cd>PLKNR</Cd>
        </ClrSysId>
      <Mmbld>777</Mmbld>
    </ClrSysMmbld>
  </FinInstnId>
  </DbtrAgt>
  <CdtTrfTxInf>
    <PmtId>
      <InstrId>777888999</InstrId>
      <EndToEndId>444555666</EndToEndId>
    </PmtId>
    <PmtTpInf>
      <SvcLvl>
        <Cd>STAN</Cd>
      </SvcLvl>
    </PmtTpInf>
    <Amt>
```

```
<InstdAmt Ccy="PLN">100</InstdAmt>
</Amt>
<CdtrAgt>
  <FinInstnId>
    <ClrSysMmbld>
      <Mmbld>888</Mmbld>
    </ClrSysMmbld>
  </FinInstnId>
</CdtrAgt>
<Cdtr>
  <Nm>Kowalski Jan</Nm>
  <PstlAdr>
    <AdrLine>Street 2</AdrLine>
    <AdrLine>00-101 City</AdrLine>
  </PstlAdr>
</Cdtr>
<CdtrAcct>
  <Id>
    <Othr>
      <Id>70167010561715920040171918</Id>
    </Othr>
  </Id>
</CdtrAcct>
<Purp>
  <Prtry>PLKR</Prtry>
</Purp>
<RmtInf>
  <Ustrd>Description Example</Ustrd>
</RmtInf>
</CdtTrfTxInf>
</PmtInf>
</CstmrCdtTrfInitn>
</soapenv:Body>
</soapenv:Envelope>
```