

TECHNICAL REQUIREMENTS FOR THE CLIENT API OF PPF BANKA A.S.

Contents:

1	INTRODUCTORY PROVISIONS	2
2	BASIC PRINCIPLES	2
3	CONFIGURING INBOUND ACCESS	2
4	CERTIFICATES FOR SECURE COMMUNICATION	3
5	THE ACTIVE PART – SUBMISSION OF PAYMENT ORDERS	3
5.1	Internal time protection	3
5.2	Allowed Characters	3
5.2.1	Allowed Characters for Domestic Orders	3
5.2.2	Allowed Characters for Foreign Orders.....	4
5.2.3	Allowed Characters for SEPA Orders	4
5.3	Payment Order statuses	4
5.3.1	List of statuses	4
5.3.2	Logic of status assignment for instant payments	5
5.4	List of available currencies	5
6	LIST OF AVAILABLE CODE LISTS	5

1 Introductory provisions

The conditions for using the Client API are set out in the *Business Conditions of PPF banka a.s. for Client API* (hereinafter “the SBC for API”), in the contract documents for Client API (hereinafter “API”), in the *General Business Conditions of PPF banka a.s.* (hereinafter “the GBC”), and in these Technical Requirements for the API.

Where the text of the Technical Requirements for the Client API contains terms, abbreviations or phrases beginning with a capital letter, these shall have the meaning stipulated in the article “Definition of terms and Interpretation rules” of the GBC, and/or *Business Conditions of PPF banka a.s. for Payments* (hereinafter “SBC for PS”), and/or SBC for API, or the meaning specified in the individual provisions of the GBC, the SBC for PS and/or the SBC for API and/or these Technical Requirements for the Client API. The current versions of the SBC for PS, the SBC for API, the GBC, and the Technical Requirements for the Client API are available at <https://www.ppfbanka.cz/en>.

User support for the API is provided by Customer Service. The contact details of the Customer Service and its Business hours are available on the [Bank's website](#).

2 Basic principles

The API for secured communication with the Client (hereinafter “Client API” or “API”) is a separate channel of Electronic Banking, which enables machine communication with the Bank concerning Account information (the Passive Part) and Payments (the Active Part).

API services are provided using the Rest API standard. The prescribed format is JSON (JavaScript Object Notation).

When a Client and the Bank communicate, secured connection is created with the help of SSL. For setting up secured communication, the Client certificate issued by the Bank must be used.

API communication is always one-step communication, i.e. only a one-off session is created. Thus, for every connection with the Bank a new secured communication session has to be opened using the Client certificate.

For all services provided by the API, the parameter Client identifier (ClientId), which the Bank defines and transfers to the Client as part of registering the client certificate for the API, may be provided in the heading of the query.

In payment services, Payment Orders are authorised by the Signing certificate. The Payment Order (configured in the JSON format) is signed by this Signing certificate and the resulting encrypted text is transmitted in the query in a separate heading (X-Content-Signature) together with the heading of the User identifier (userId, generated by the Bank) and the Payment Order in the JSON format in the body of the query.

The transmission of an identifier (of the account or of the Payment Order, as applicable) as part of the URL is required for providing information about the balance in the Account, Payment Transactions cleared and not yet cleared to the Account, and the status of submitted Payment Orders.

The filter of information about cleared and not yet cleared Payment Transactions is transferred as a set of parameters in the URL (the set of parameters following the question mark).

Documentation for the API solution (swagger including an overview of changes, a server for accessing the Client API, etc.), a sandbox for tests of the functionality of the Client API, as well as operational information (especially information about planned shutdowns) can be found on the [Bank's Website](#).

3 Configuring inbound access

For the Client API to be used, inbound access must be configured for the IP addresses from which the Client will access the Client API. The API Administrator must report these IP addresses via a message in Internet Banking (“IB”).

If the IP addresses from which the Client will access the Client API are changed, the API Administrator must report these changes to the Bank in good time to avoid any interruption in service provision.

The Bank will automatically refuse any attempt to call the Client API from IP addresses other than those reported.

4 Certificates for secure communication

The following certificates issued by the Bank are used for secure communication via the Client API:

- Client Certificate – this is always issued, and is managed by the API Administrator;
- Signing Certificate
 - this is issued only if the Client will also use the active part of the API and send Payment Orders to the Bank;
 - it is managed by the API User.

Client Certificates and Signing Certificates (“Certificates”) are issued by the Bank’s own certification authority, so this must be taken into account during communications (e.g. by disabling reverse authentication).

For the purpose of generating Certificate requests the Bank sends strings to the API Administrator via a message in IB.

Certificate requests can be generated with these strings in Linux or Windows using the KeyStore application (the generation procedure is described in a separate document *Generating certificates for the Client API in Windows* on [the Bank’s website](#)).

The API Administrator must generate requests (.csr files) with the exact name stated in the message sent by the Bank in IB. The .key, jks, or similar files that are generated are to be saved by the API Administrator for further use, and are not sent to the Bank. The Bank will then send the generated Certificates (.cer files) to the API Administrator by another message in IB.

Certificates are valid for one year. New Certificates can be generated and stored in the Client’s systems before existing Certificates expire. After the existing Certificates expire, these new Certificates are automatically used for communication without the need to interrupt communication via the Client API.

5 The active part – submission of Payment Orders

5.1 Internal time protection

The internal time protection of 5 seconds is set in the system for obtaining a Payment Order. If the sent Payment Order is not created in the Bank’s system within this time limit, the entire request for the creation of a Payment Order is cancelled and it is possible to send it again.

5.2 Allowed Characters

If disallowed characters are entered in the Payment Orders, the individual items may be rejected due to formal errors after sending to the Bank or after transmission to other subjects (CNB, foreign banks).

Therefore we recommend to modify the software which generates Payment Orders sending via Client’s API so that all disallowed characters cannot be entered at all.

5.2.1 Allowed Characters for Domestic Orders

Characters that are allowed in the CERTIS system can only be entered in the Domestic Order¹:

a b c d e f g h i j k l m n o p q r s t u v w x y z á ä ç d é ë í ľ ň ó ô ö ř ř š ť ú ů ů ý ž

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z Á Ä Ç Ď ě ě Ĺ Ń Ó Ô Ö Ř Ř Š Ť Ú Ů Ů Ý Ž

0 1 2 3 4 5 6 7 8 9

/ - ? : () . , ‘ + ! “ # \$ % & * ; < = > @ [\] ^ ` { | } ~ \$

space (must not be at the beginning of a line)

¹ <https://www.cnb.cz/en/payments/certis/>

5.2.2 Allowed Characters for Foreign Orders

Characters that are allowed in the [SWIFT](#) messages can only be entered in the Foreign Order²:

a b c d e f g h i j k l m n o p q r s t u v w x y z

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

0 1 2 3 4 5 6 7 8 9

/ - ? : () . , ' +

space (must not be at the beginning of a line)

5.2.3 Allowed Characters for SEPA Orders

Characters that can only be entered in the SEPA Order³:

a b c d e f g h i j k l m n o p q r s t u v w x y z

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

0 1 2 3 4 5 6 7 8 9

/ - ? : () . , ' +

space (must not be at the beginning of a line)

5.3 Payment Order statuses

5.3.1 List of statuses

ID	Code of the status	Final status	Description of the status
1	EDIT	NE	The Payment order is edited.
25	ERROR	YES – ERROR	The Payment order was rejected in the Bank.
35	DELETED	YES	The Payment order was deleted by the user (before signing).
39	ERROR_1	YES – ERROR	The Payment order was rejected due to formal errors – will not be executed,
40	AFS_CHECK	NO	Anti-fraud system check is in progress.
	NEW	NO	The Payment order was created.
	PASSING	NO	The Payment order has been transmitted to the Bank's systems.
	VERIFIED	NO	The Payment order was successfully signed.
43	ERROR_2	YES – ERROR	The Payment order was rejected by the Bank due to technical error.
46	PASSED	NO	The Payment order was passed to the Bank.
48	WAITDONE	NO	The Payment order is waiting to be completed.
55	WAITMATUR	NO	The Payment order is waiting for the Maturity Date.
56	WAITFUNDS	NO	The Payment order is waiting for sufficient funds in the account.
59	WAITAUTH	NO	The Payment order is waiting for authorisation in the Bank.
60	ACCOK	NO	The Payment order has been executed and is awaiting posting.

² <https://www.swift.com/>

³ <https://www.europeanpaymentscouncil.eu/document-library/guidance-documents/sepa-requirements-extended-character-set-unicode-subset-best>

ID	Code of the status	Final status	Description of the status
61	ACCOK2	NO	The Payment order was successfully submitted to the CNB.
65	CANCELLED	YES	The Payment order was cancelled at the Client's request.
89	ERROR_3	YES – ERROR	The Payment order was rejected in the Bank.
90	DONE	YES	The Payment order has been executed and posted.

5.3.2 Logic of status assignment for instant payments

For instant payments in CZK sent to other banks on Business Days from approximately 19:30 to 5:15 and outside Business Days, instead of status 46 – PASSED, status 60 – ACCOCK is assigned to the Payment Order.

After the above times have elapsed, status 90 – DONE will be assigned to the instant payment.

5.4 List of available currencies

Name	Designation	Name	Designation
Australian dollar	AUD	Bulgarian lev	BGN
Canadian dollar	CAD	Chinese yuan	CNY
Czech crown	CZK	Euro	EUR
British pound	GBP	Hong Kong dollar	HKD
Hungarian forint	HUF	Swiss franc	CHF
Indonesian rupiah	IDR	Indian rupee	INR
Japanese yen	JPY	Kazakhstan tenge	KZT
Norwegian crown	NOK	Polish zloty	PLN
Romanian new leu	RON	Serbian dinar	RSD
Russian rouble	RUB	Swedish crown	SEK
Turkish lira	TRY	US dollar	USD

6 List of available code lists

- Available code lists:

- o a list of banks with information on whether the bank is involved in the instant payment scheme
- o a list of available currencies
- o a list of transaction type codes (DPO, FPO, IPO, etc.)
- o a list of transaction types in detail (Card transaction - POS, Domestic transaction – outgoing, ..)
- o a list of payment order statuses

- Response example:

```
{
  "codebooks": [
    {
      "name": "BANKS",
      "data": [
        {
          "code": "0100",
          "texts": {
            "en": "KOMERCNI BANKA A.S.",
            "cs": "KOMERCNI BANKA A.S."
          },
          "details": {"instant": "1"}
        },
        {
          "code": "2700",
          "texts": {
            "en": "UniCredit Bank Czech Republic and Slovakia, a.s.",
            "cs": "UniCredit Bank Czech Republic and Slovakia, a.s."
          }
        }
      ]
    }
  ]
}
```

```

    "details": {}
  },
  "name": "CURRENCIES",
  "data": [
    {
      "code": "EUR",
      "texts": {
        "en": "Euro",
        "cs": "Euro"
      }
    },
    "details": {}
  ],
  "name": "ACCOUNTMOVEMENTGROUPS",
  "data": [
    {
      "code": "DPO",
      "texts": {
        "en": "Domestic",
        "cs": "Tuzemská"
      }
    },
    "details": {}
  ],
  "name": "ACCOUNTMOVEMENTTYPES",
  "data": [
    {
      "code": "303",
      "texts": {
        "en": "Card transaction - POS",
        "cs": "Transakce PK - POS"
      }
    },
    "details": {}
  ],
  "name": "CLIENTTRANSACTIONSTATUSTYPES",
  "data": [
    {
      "code": "60",
      "texts": {
        "en": "ACCEPTED. Accomplished.",
        "cs": "ACCEPTED. Provedeno."
      }
    }
  ]

```