



ČESKÁCZECH
BANKOVNÍBANKING
ASOCIACEASSOCIATION

Banking Activities Standards

Standard ČBA

Czech Standard for Open Banking

Version 7.0

Public from: 27. 09. 2023

Valid from: 27. 03. 2024

Changes

Date	Note	Author
04.11.2022	Basic document v.07.0	ČS Petr Michalík
21.11.2022	New authorization_redirect_uris element in the dynamic application registration API. See chapter 1.4.1 (1.4.1.2)	ČS Petr Michalík
21.11.2022	New error code in authorization flow when entering an unregistered redirect_uri. See chapters 3.2.10, 3.2.19 and 3.2.26	ČS Petr Michalík
21.11.2022	New types of callback error states for better assessment of failed logins. See chapter 1.4.3	ČS Petr Michalík
22.11.2022	New element in the overview of accounts, determining if the authenticated person is the owner of the account. See chapters 3.1.3 and 5.2.2	ČS Petr Michalík
3.1.2023	Domestic Direct Debits See chapters 3.2.39 – 3.2.45	ČS Petr Michalík ČS Michaela Kozarovská
15.2.2023	New API version for payment initiation with one SCA See chapter 3.2.30	Tomáš Rosa ČS Petr Michalík
20.6.2023	The new division of scopes for restricting access to services. See chapter 1.4.1.1	ČS Petr Michalík
25.6.2023	Recommendations for the use of universal and app links in bank mobile authentication applications see chapter 1.7.	ČS Petr Michalík
18.9.2023	Added alternative flow in case of payment with one SCA, see chapter 3.2.30	ČS Petr Michalík

Table of Contents

TABLE OF CONTENTS	3
1 COBS CONTENT	17
1.1 A GENERAL DESCRIPTION OF STANDARD CONTENT	17
1.2 TECHNICAL DESCRIPTION	17
1.2.1 <i>Transport protocol</i>	17
1.2.2 <i>Interface architecture</i>	17
1.2.3 <i>Data entry format</i>	17
1.2.4 <i>Type of transmission media</i>	18
1.2.5 <i>Coding</i>	18
1.2.6 <i>Localization</i>	19
1.2.7 <i>HTTP methods used in API and their specifications</i>	19
1.2.8 <i>Use of filtering, sorting and pagination of API elements</i>	20
1.2.9 <i>Versioning of API</i>	24
1.2.10 <i>HTTP status codes used and addressing errors in API</i>	24
1.2.11 <i>Authentication and Request Authorization (OAuth2)</i>	27
1.2.12 <i>Security standard</i>	28
1.3 FLOW IN THE PROCESS OF CLIENT ENROLMENT TO THE TPP APPLICATION	28
1.3.1 <i>Individual flow steps</i>	28
1.4 AN OVERVIEW OF RESOURCES TO ENSURE SECURITY FLOW STANDARD	32
1.4.1 <i>Optional registration resource issued by the Bank</i>	32
1.4.2 <i>Authentication resource issued by the Bank</i>	48
1.4.3 <i>1. Authentication resource</i>	48
1.4.4 <i>2.a Get token resource</i>	54
1.4.5 <i>2.b Renewal of access token</i>	56
1.4.6 <i>2.c Token introspection</i>	58
1.4.7 <i>3. Token invalidation</i>	61
1.4.8 <i>Error codes used in the enrolment</i>	62
1.4.9 <i>TPP authentication</i>	63
1.4.10 <i>Communication security</i>	64

1.5	COMMON HTTP HEADERS.....	64
1.5.1	<i>Request headers</i>	64
1.5.2	<i>Response headers</i>	66
1.6	CONSENT MANAGEMENT.....	67
1.6.1	<i>Consent flow</i>	67
1.6.2	<i>Get list of consents (GET /my/consents)</i>	69
1.6.3	<i>Consent delete (DELETE /my/consents/{consentId})</i>	71
1.7	APP2APP COMMUNICATION.....	73
1.7.1	<i>An example of an existing flow on a mobile device</i>	73
1.7.2	<i>Example flow using universal and app links</i>	74
1.7.3	<i>Description of interactions when using universal and app link flow</i>	74
1.8	NOMENCLATURE AND BASIC CONCEPTS IDENTICAL ACROSS COBS.....	76
1.9	HEALTH CHECK AND PLANNED OUTAGES GENERIC RESOURCES	77
1.9.1	<i>Health Check</i>	77
1.9.2	<i>Example of request and response for Health Check on AIS API</i>	80
1.9.3	<i>Planned Outages (GET /my/plannedOutages)</i>	82
1.9.4	<i>Example of request and response for Planned Outages</i>	85
1.10	RECOMMENDATIONS FOR SANDBOX PSD2 ASPSP SOLUTION FEATURES.....	87
1.10.1	<i>General features of the Sandbox</i>	87
1.10.2	<i>Individual functions of PSD2 Sandbox and recommendations for their features</i>	88
2	STANDARD GOVERNANCE	89
3	SERVICES DEFINED WITHIN THE STANDARD	89
3.1	API ACCOUNT INFORMATION	90
3.1.1	<i>How to read API account information</i>	90
3.1.2	<i>List of API resources Account Information</i>	91
3.1.3	<i>List of client payment accounts (GET /my/accounts{?size,page,sort,order})</i>	92
3.1.4	<i>Account balance (GET /my/accounts/{id}/balance{?currency})</i>	96
3.1.5	<i>Overview of transactions (GET /my/accounts/{id}/transactions{?fromDate,toDate,currency,size,page,sort,order})</i>	100
3.1.6	<i>Overview of awaiting transactions (GET /my/accounts/{id}/payments/awaiting{?size,page,sort,order})</i>	154
3.1.7	<i>List of client's standing orders (GET /my/standingorders{?size,page,sort,order})</i>	158

3.1.8	Standing order detail (GET /my/standingorders/{transactionIdentification})	162
3.2	API PAYMENT INITIATION	170
3.2.1	How to read API Payment Initiation	170
3.2.2	List of API Payment Initiation resources	171
3.2.3	Query for Balance Check (POST /my/payments/balanceCheck).....	173
3.2.4	New payment - payment initiation (POST /my/payments)	177
3.2.5	Status of entered/initiated payment (GET /payments/{paymentId}/status)	198
3.2.6	Info on entered/initiated payment (GET /my/payments/{paymentId})	201
3.2.7	Delete an entered payment (DELETE /my/payments/{paymentId}).....	202
3.2.8	Generating the authorization ID (POST /my/payments/{paymentId}/sign).....	204
3.2.9	Step I. Payment authorization detail (GET /my/payments/{paymentId}/sign/{signId}).....	207
3.2.10	Step II. Payment authorization initiation - specific for each bank (POST /my/payments/{paymentId}/sign/{signId})	209
3.2.11	Step III. Payment authorization initiation – specific for each bank (PUT /my/payments/{paymentId}/sign/{signId})	212
3.2.12	Standing orders - recurring payments	215
3.2.13	New standing order - instruction initiation (POST /my/standingorders).....	215
3.2.14	Standing order detail (GET /my/standingorders/{transactionIdentification})	225
3.2.15	Standing order status (GET /my/standingorders/{transactionIdentification}/status	233
3.2.16	Standing order delete (DELETE /my/standingorders/{transactionIdentification})	235
3.2.17	Existing standing order change (PUT /my/standingorders/{transactionIdentification}).....	237
3.2.18	Standing order authorization Step I (GET /my/standingorders/{transactionIdentification}/sign/{signId})	247
3.2.19	Standing order authorization Step II (POST /my/standingorders/{transactionIdentification}/sign/{signId})	249
3.2.20	Standing order authorization Step III (PUT /my/standingorders/{transactionIdentification}/sign/{signId})	252
3.2.21	Batch payment - bulk authorization	254
3.2.22	New batch payment - instruction initiation (POST /my/batchpayments)	254
3.2.23	Batch payment delete (DELETE /my/batchpayments/{transactionIdentification}{?allInstructions})	258
3.2.24	Batch payment status (GET /my/batchpayments /{transactionIdentification}/status	259

3.2.25	Batch payment authorization Step I (GET /my/batchpayments/{transactionIdentification}/sign/{signId})	261
3.2.26	Batch payment authorization Step II (POST /my/batchpayments/{transactionIdentification}/sign/{signId})	263
3.2.27	Batch payments authorization Step III (PUT /my/batchpayments/{transactionIdentification}/sign/{signId})	266
3.2.28	Get list of authorizations (GET /my/authorizations).....	268
3.2.29	List of client's awaiting payments (GET /my/payments/awaiting{?size,page,sort,order})	270
3.2.30	Payment initialization with one SCA	274
3.2.31	PayLater payment (loan)	283
3.2.32	PayLater payment (loan) sequence diagram	283
3.2.33	Overview of unsecured loans initiated via TPP (GET/my/loans/{clientID})	284
3.2.34	Loan offers (GET/my/loanoffers{?loanPeriod,amount,currency})	287
3.2.35	Loan application (POST/my/loans)	290
3.2.36	Loan application status (GET/my/loans/{loanApplicationID}/status)	292
3.2.37	Initiation of drawing a loan (POST /my/loans/{loanApplicationID}/draw)	295
3.2.38	Loan application cancellation (DELETE/my/loans/{loanApplicationID})	296
3.2.39	Domestic direct debit consents – consent with payments initiated by the creditor	298
3.2.40	Domestic direct debit consent - instruction initiation (POST /my/accounts/{id}/directdebits)	298
3.2.41	Domestic direct debit Consent detail (GET /my/accounts/{id}/directdebits/{mandateIdentification})	301
3.2.42	List of client's domestic direct debits Consents (GET /my/accounts/{id}/directdebits{?size,page,sort,order})	303
3.2.43	Change of existing Domestic direct debit Consent (PUT /my/accounts/{id}/directdebits/{id}/) 305	
3.2.44	Domestic direct debit Consent delete (DELETE /my/accounts/{id}/directdebits/{id}/)	308
3.2.45	Direct debit authorization Step I (GET my/accounts/{id}/directdebits/{mandateIdentification}/sign/{signId})	309
3.2.46	Direct debit authorization Step II (POST /my/accounts/{id}/directdebits/{mandateIdentification}/sign/{signId})	310
3.2.47	Direct debit authorization Step III (PUT /my/accounts/{id}/directdebits/{mandateIdentification}/sign/{signId})	312
3.3	API BALANCE CHECK.....	314
3.3.1	How to read API Balance Check	314

3.3.2	List of API Balance Check resources	315
3.3.3	Query for Balance Check (POST /accounts/balanceCheck).....	316
3.4	KNOW YOUR CUSTOMER AND USER INFO SERVICE	320
3.4.1	User info service.....	320
3.4.2	Example of request and response for Query user info service.....	326
3.4.3	User profile.....	330
3.4.4	Example of request and response for Query user profile service.....	335
3.5	API ACCOUNT STATEMENT	337
3.5.1	How to read API Account Statement	337
3.5.2	List of API resources Account Statement	338
3.5.3	List of available account statements (GET /my/accounts/{id}/statements{?year,month,availability }).....	339
3.5.4	Get account statement download Url (GET /my/accounts/{id}/statement/download{?accountStatementId})	346
3.5.5	Code lists for account statements.....	355
4	DESCRIPTION OF ELEMENTS OF MESSAGES FOR SERVICES PAYMENT INITIATION, ACCOUNT INFORMATION AND BALANCE CHECK	356
4.1	AMOUNT	356
4.1.1	InstructedAmount (Amount and currency in the instruction).....	357
4.1.2	EquivalentAmount (Equivalent amount and currency).....	358
4.2	AUTHENTICATIONMETHOD (AUTHENTICATION METHOD)	359
4.3	BANKTRANSACTIONCODE (CODE OF BANK TRANSACTION)	360
4.4	BOOKINGDATE (DATE OF PROCESSING).....	364
4.5	CARD (CARD)	365
4.5.1	cardholderName (Card holder name).....	366
4.5.2	maskedPan (Masked card number).....	366
4.6	CODE (THE BALANCE TYPE IN THE ACCOUNT BALANCE RESPONSE)	367
4.7	CREDITDEBITINDICATOR (INDICATION OF DEBIT/CREDIT PAYMENT)	368
4.8	CREDITOR (PAYEE)	368
4.8.1	name (Name).....	369
4.8.2	postalAddress (Payee postal address)	370
4.9	CREDITORACCOUNT (PAYEE ACCOUNT)	374

4.9.1	<i>identification (Identification)</i>	375
4.9.2	<i>currency (Currency)</i>	378
4.10	CREDITORAGENT (PAYEE BANK)	379
4.10.1	<i>financialInstitutionIdentification (Financial institution identification)</i>	380
4.11	DEBTOR (PAYER)	388
4.11.1	<i>name (Name)</i>	388
4.11.2	<i>postalAddress (Postal address of the original payer)</i>	389
4.12	DEBTORACCOUNT (PAYER ACCOUNT)	392
4.12.1	<i>identification (Identification)</i>	393
4.12.2	<i>currency (Currency)</i>	396
4.13	SEPA IDENTIFICATION (IDENTIFICATION)	397
4.13.1	<i>organisationIdentification (Organization identification)</i>	398
4.13.2	<i>privateIdentification (Private individual identification)</i>	403
4.14	ENTRYDETAILS (ENTRY DETAILS)	407
4.14.1	<i>transactionDetails (Payment details)</i>	408
4.15	ENTRYREFERENCE (PAYMENT NUMBER)	424
4.16	EXCHANGEIDENTIFICATION (IDENTIFIER ASSIGNED BY THIRD PARTY)	424
4.17	EXCHANGERATEINFORMATION (CONTRACTUAL EXCHANGE RATE)	425
4.17.1	<i>exchangeRate (Agreed Exchange rate)</i>	425
4.17.2	<i>rateType (Type of Agreed Exchange rate)</i>	425
4.17.3	<i>contractIdentification (Identifier of the use of Agreed Exchange rate)</i>	426
4.18	CHARGESACCOUNT (ACCOUNT FOR FEES)	426
4.18.1	<i>identification (Identification of account number for fees)</i>	426
4.18.2	<i>currency (Fee account currency)</i>	427
4.19	CHARGEBEARER (FEE PAYER)	427
4.20	INTERMEDIARYAGENT1 (INTERMEDIARY BANK 1)	428
4.20.1	<i>financialInstitutionIdentification (Financial institution identification)</i>	429
4.21	INSTRUCTIONFORNEXTAGENT (INSTRUCTION FOR ANOTHER BANK)	433
4.22	MERCHANT (MERCHANT)	434
4.22.1	<i>identification (Identification)</i>	435
4.22.2	<i>type (Entity type)</i>	436

4.22.3	<i>shortName (Entity name)</i>	436
4.22.4	<i>commonName (Entity name)</i>	437
4.22.5	<i>address (Address)</i>	437
4.22.6	<i>countryCode (Country code)</i>	438
4.22.7	<i>merchantCategoryCode (Category code)</i>	438
4.23	PAYMENTIDENTIFICATION (PAYMENT IDENTIFICATION)	439
4.23.1	<i>instructionIdentification (Instruction identification)</i>	439
4.23.2	<i>endToEndIdentification (End To End identification)</i>	440
4.23.3	<i>transactionIdentification (Transaction identification)</i>	440
4.24	PAYMENTTYPEINFORMATION (INFORMATION ON PAYMENT TYPE).....	441
4.24.1	<i>instructionPriority (Instruction priority)</i>	442
4.24.2	<i>serviceLevel (Service level)</i>	443
4.24.3	<i>categoryPurpose (Payment purpose category)</i>	444
4.25	INITIATINGPARTY (INITIATING PARTY).....	445
4.25.1	<i>name</i>	446
4.25.2	<i>postalAddress</i>	446
4.26	PURPOSE (PAYMENT PURPOSE)	446
4.26.1	<i>code (Code)</i>	447
4.26.2	<i>proprietary (Free format)</i>	447
4.27	REMITTANCEINFORMATION (PAYMENT INFORMATION)	447
4.27.1	<i>unstructured (Unstructured message for payee)</i>	448
4.27.2	<i>structured (Structured message for payee – variable, specific and constant symbol)</i>	449
4.27.3	<i>additionalRemittanceInformation (Unstructured message for payer)</i>	450
4.28	REQUESTEDEXECUTIONDATE (REQUIRED PAYMENT EXECUTION DATE)	451
4.29	REVERSALINDICATOR (CANCELLATION).....	451
4.30	STANDINGORDER (RECURRING PAYMENT INSTRUCTION SPECIFICATION)	452
4.30.1	<i>execution (Features of execution)</i>	454
4.30.2	<i>validity (restricting the standing order validity)</i>	458
4.30.3	<i>exceptions (exceptions to executing)</i>	461
4.31	STATUS (STATUS)	464
4.32	TRANSACTIONDETAILS (TRANSACTION DETAILS)	464

4.32.1	<i>currency (Currency)</i>	465
4.32.2	<i>totalAmount (Total amount)</i>	466
4.33	ULTIMATECREDITOR (FINAL PAYEE)	466
4.33.1	<i>name (Name)</i>	467
4.33.2	<i>postalAddress (Postal address)</i>	468
4.33.3	<i>identification (Identification)</i>	471
4.34	ULTIMATEDEBTOR (ORIGINAL PAYEE)	472
4.34.1	<i>name (Name of original payee)</i>	473
4.34.2	<i>postalAddress (Postal address of original payer)</i>	473
4.34.3	<i>identification (Identification)</i>	477
4.35	VALUEDATE (DUE DATE)	477
4.35.1	<i>date (Date)</i>	478
5	EXAMPLES OF REQUEST AND RESPONSE FOR INDIVIDUAL MESSAGES	480
5.1	EXAMPLE OF REQUEST AND RESPONSE FOR QUERY FOR BALANCE CHECK AT ACCOUNT.	480
5.1.1	<i>Example of request</i>	480
5.1.2	<i>Example of correct response</i>	481
5.1.3	<i>Example of response to wrong call</i>	481
5.2	EXAMPLE OF REQUEST AND RESPONSE FOR QUERY FOR LIST OF ACCOUNTS OF BANK'S CLIENT	482
5.2.1	<i>Example of request</i>	482
5.2.2	<i>Example of correct response</i>	483
5.2.3	<i>Example of response to wrong call</i>	484
5.3	EXAMPLE OF REQUEST AND RESPONSE FOR QUERY FOR BALANCE IN A SPECIFIC ACCOUNT OF BANK'S CLIENT	485
5.3.1	<i>Example of request</i>	485
5.3.2	<i>Example of correct response</i>	485
5.3.3	<i>Example of response to wrong call</i>	486
5.4	EXAMPLE OF REQUEST AND RESPONSE FOR QUERY FOR A TRANSACTION OVERVIEW IN A SPECIFIC ACCOUNT OF BANK'S CLIENT	487
5.4.1	<i>Example of request</i>	487
5.4.2	<i>Example of correct response</i>	488
5.4.3	<i>Example of response to wrong call</i>	498

5.5	EXAMPLE OF REQUEST AND RESPONSE FOR QUERY FOR A AWAITING TRANSACTION OVERVIEW IN A SPECIFIC ACCOUNT OF BANK'S CLIENT	500
5.5.1	<i>Example of request</i>	500
5.5.2	<i>Example of correct response</i>	500
5.5.3	<i>Example of response to wrong call</i>	502
5.6	EXAMPLE OF REQUEST AND RESPONSE AFTER ENTERING A NEW PAYMENT	504
5.6.1	<i>Example of request Domestic payment</i>	504
5.6.2	<i>Example of request SEPA payment</i>	505
5.6.3	<i>Example of request Foreign payment within EEA</i>	508
5.6.4	<i>Example of request Foreign payment outside EEA</i>	510
5.6.5	<i>Example of correct response</i>	512
5.6.6	<i>Example of responses to wrong call</i>	514
5.7	EXAMPLE OF REQUEST AND RESPONSE FOR QUERY FOR STATUS OF ENTERED/INITIATED PAYMENT	515
5.7.1	<i>Example of request</i>	515
5.7.2	<i>Example of correct response</i>	516
5.7.3	<i>Example of response to wrong call</i>	516
5.8	EXAMPLE OF REQUEST AND RESPONSE FOR QUERY FOR INFO ON ENTERED/INITIATED PAYMENT	517
5.8.1	<i>Example of request</i>	517
5.8.2	<i>Example of correct response</i>	517
5.8.3	<i>Example of response to wrong call</i>	519
5.9	EXAMPLE OF REQUEST AND RESPONSE TO DELETE UNAUTHORISED AND AUTHORISED PAYMENTS	519
5.9.1	<i>Example of request</i>	519
5.9.2	<i>Example of correct response</i>	520
5.9.3	<i>Example of response to wrong call</i>	520
5.10	EXAMPLE OF REQUEST AND RESPONSE TO GENERATE PAYMENT AUTHORIZATION ID	521
5.10.1	<i>Example of request</i>	521
5.10.2	<i>Example of correct response</i>	521
5.10.3	<i>Example of response to wrong call</i>	522
5.11	EXAMPLE OF REQUEST AND RESPONSE FOR PAYMENT AUTHORIZATION - STEP I. PAYMENT AUTHORIZATION DETAIL 523	
5.11.1	<i>Example of request</i>	523

5.11.2	<i>Example of correct response</i>	523
5.11.3	<i>Example of response to wrong call</i>	524
5.12	EXAMPLE OF REQUEST AND RESPONSE FOR PAYMENT AUTHORIZATION - STEP II. PAYMENT AUTHORIZATION INITIATION - SPECIFIC FOR EACH BANK	525
5.12.1	<i>Example of requests</i>	525
5.12.2	<i>Example of correct response</i>	526
5.12.3	<i>Example of response to wrong call</i>	526
5.13	EXAMPLE OF REQUEST AND RESPONSE FOR PAYMENT AUTHORIZATION - STEP III. PAYMENT AUTHORIZATION FINALIZATION - SPECIFIC FOR EACH BANK	527
5.13.1	<i>Example of request</i>	528
5.13.2	<i>Example of correct response</i>	528
5.13.3	<i>Example of response to wrong call</i>	529
5.14	EXAMPLE OF REQUEST AND RESPONSE FOR LIST OF CLIENT’S STANDING ORDERS.....	529
5.14.1	<i>Example of request</i>	529
5.14.2	<i>Example of correct response</i>	529
5.14.3	<i>Example of response to wrong call</i>	531
5.15	EXAMPLE OF REQUEST AND RESPONSE AFTER ENTERING A NEW STANDING ORDER.....	531
5.15.1	<i>Example of request standing order initiation for Domestic payment</i>	531
5.15.2	<i>Example of correct response</i>	534
5.15.3	<i>Example of response to wrong call</i>	536
5.16	EXAMPLE OF REQUEST AND RESPONSE FOR GET DETAIL ABOUT STANDING ORDER (AISP AND ALSO PISP)	537
5.16.1	<i>Example of request detail of standing order</i>	537
5.16.2	<i>Example of correct response</i>	537
5.16.3	<i>Example of response to wrong call</i>	539
5.17	EXAMPLE OF REQUEST AND RESPONSE FOR GET STATUS OF STANDING ORDER.....	540
5.17.1	<i>Example of request</i>	540
5.17.2	<i>Example of correct response</i>	540
5.17.3	<i>Example of response to wrong call</i>	541
5.18	EXAMPLE OF REQUEST AND RESPONSE TO DELETE UNAUTHORISED AND AUTHORISED STANDING ORDER	542
5.18.1	<i>Example of request</i>	542
5.18.2	<i>Example of correct response</i>	542

5.18.3	<i>Example of response to wrong call</i>	543
5.19	EXAMPLE OF REQUEST AND RESPONSE TO CHANGE EXISTING STANDING ORDER	543
5.19.1	<i>Example of request standing order change (Domestic payment)</i>	543
5.19.2	<i>Example of correct response</i>	545
5.19.3	<i>Example of response to wrong call</i>	547
5.20	EXAMPLE OF REQUEST AND RESPONSE STEP I (GET AUTHORIZATION) OF STANDING ORDER	549
5.20.1	<i>Example of request</i>	549
5.20.2	<i>Example of correct response</i>	549
5.20.3	<i>Example of response to wrong call</i>	550
5.21	EXAMPLE OF REQUEST AND RESPONSE STEP II (TRIGGERING AUTHORIZATION) OF STANDING ORDER.....	550
5.21.1	<i>Example of requests</i>	550
5.21.2	<i>Example of correct response</i>	551
5.21.3	<i>Example of response to wrong call</i>	552
5.22	EXAMPLE OF REQUEST AND RESPONSE STEP III (VALIDATE/END AUTHORIZATION) OF STANDING ORDER	553
5.22.1	<i>Example of request</i>	553
5.22.2	<i>Example of correct response</i>	554
5.22.3	<i>Example of response to wrong call</i>	554
5.23	EXAMPLE OF REQUEST AND RESPONSE AFTER ENTERING A BATCH PAYMENT	554
5.23.1	<i>Example of request batch payment initiation</i>	555
5.23.2	<i>Example of correct response</i>	555
5.23.3	<i>Example of response to wrong call</i>	556
5.24	EXAMPLE OF REQUEST AND RESPONSE FOR BATCH PAYMENT DELETE.....	557
5.24.1	<i>Example of request</i>	557
5.24.2	<i>Example of correct response</i>	557
5.24.3	<i>Example of response to wrong call</i>	557
5.25	EXAMPLE OF REQUEST AND RESPONSE FOR BATCH PAYMENT STATUS	558
5.25.1	<i>Example of request</i>	558
5.25.2	<i>Example of correct response</i>	558
5.25.3	<i>Example of response to wrong call</i>	559
5.26	EXAMPLE OF REQUEST AND RESPONSE TO BATCH PAYMENT AUTHORIZATION STEP I	560
5.26.1	<i>Example of request</i>	560

5.26.2	<i>Example of correct response</i>	560
5.26.3	<i>Example of response to wrong call</i>	561
5.27	EXAMPLE OF REQUEST AND RESPONSE TO BATCH PAYMENT AUTHORIZATION STEP II.....	561
5.27.1	<i>Example of requests</i>	561
5.27.2	<i>Example of correct response</i>	562
5.27.3	<i>Example of response to wrong call</i>	563
5.28	EXAPMLE OF REQUEST AND RESPONSE FOR BATCH PAYMENT AUTHORIZATION STEP III.....	564
5.28.1	<i>Example of request</i>	564
5.28.2	<i>Example of correct response</i>	564
5.28.3	<i>Example of response to wrong call</i>	565
5.29	EXAMPLE OF REQUEST AND RESPONSE FOR LIST OF CONSENTS.....	565
5.29.1	<i>Example of request for list of consents</i>	565
5.29.2	<i>Example of correct response</i>	566
5.29.3	<i>Example of response to wrong call</i>	567
5.30	EXAMPLE OF REQUEST AND RESPONSE FOR DELETE AN CONSENT.....	567
5.30.1	<i>Example of request for deleting of consents</i>	567
5.30.2	<i>Example of correct response</i>	568
5.30.3	<i>Example of response to wrong call</i>	568
5.31	EXAMPLE OF REQUEST AND RESPONSE FOR LIST OF AUTHORIZATIONS.....	568
5.31.1	<i>Example of request for list of authorizations</i>	568
5.31.2	<i>Example of correct response</i>	569
5.31.3	<i>Example of response to wrong call</i>	570
5.32	EXAMPLE OF REQUEST AND RESPONSE FOR QUERY FOR LIST OF CLIENT'S AWAITING PAYMENTS.....	571
5.32.1	<i>Example of request</i>	571
5.32.2	<i>Example of correct response</i>	571
5.32.3	<i>Example of response to wrong call</i>	573
5.33	EXAMPLE OF REQUEST AND RESPONSE ROS REGISTRATION.....	575
5.33.1	<i>Example of payment response with authorization object</i>	575
5.33.2	<i>Example of request</i>	577
5.33.3	<i>Example of correct response</i>	578
5.33.4	<i>Example of correct IDToken</i>	579

5.34	EXAMPLE OF REQUEST AND RESPONSE OVERVIEW OF UNSECURED LOANS INITIATED VIA TPP	579
5.34.1	<i>Example of request</i>	579
5.34.2	<i>Example of correct response</i>	580
5.35	EXAMPLE OF REQUEST AND RESPONSE LOAN OFFERS (GET/MY/LOANOFFERS{?LOANPERIOD,AMOUNT,CURRENCY})	581
5.35.1	<i>Example of request</i>	581
5.35.2	<i>Example of correct response</i>	581
5.36	EXAMPLE OF REQUEST AND RESPONSE LOAN APPLICATION (POST /MY/LOANS).....	582
5.36.1	<i>Example of request</i>	582
5.36.2	<i>Example of callback response (on specified callbackURL)</i>	583
5.37	EXAMPLE OF REQUEST AND RESPONSE LOAN APPLICATION STATUS (GET /MY/LOANS/{LOANAPPLICATIONID}/STATUS).....	583
5.37.1	<i>Example of request</i>	583
5.37.2	<i>Example of correct response</i>	584
5.38	EXAMPLE OF REQUEST AND RESPONSE INITIATION OF DRAWING A LOAN (POST /MY/LOANS/{LOANAPPLICATIONID}/DRAW)	584
5.38.1	<i>Example of request</i>	584
5.38.2	<i>Example of correct response</i>	585
5.39	EXAMPLE OF REQUEST AND RESPONSE LOAN APPLICATION CANCELLATION (DELETE/MY/LOANS/{LOANAPPLICATIONID})	585
5.39.1	<i>Example of request</i>	585
5.39.2	<i>Example of correct response</i>	586

Introduction

On 16 November 2015, the European Banking Authority issued the Revised Payment Service Directive (hereinafter: PSD2). As of 13th January 2018, PSD2 will replace former Directive on Payment Services in the Internal Market. A need for this substitution was caused by a rapid development of IT, by the birth of brand new types of payment services and by a number of overall changes. Thanks to PSD2 new subjects will be regulated in the field of electronic payments, subjects, which were not in the scope of the previous Directive. Among other things, new legislation opens banking for the so-called third parties.

PSD 2 should be accompanied by number of norms: on one hand certain regulatory technical standards (RTS) have to be adopted, on the other hand guidelines describing rules, which have to be followed in specific situations have to be issued by the EBA (European Banking Authority) too. Unfortunately, during preparation of these documents both the EBA and the European Commission caused some delays, therefore it was necessary to work only with drafts of these papers. Especially missing RTS on Strong Authentication and Secure Communication (RTS on SCA) caused troubles to all stakeholders.

The PSD 2 was implemented into the Czech legislation by the brand new No 370/2018 Coll., Act on Payment Systems, which will come into force on 13th January 2018. Suspended are only those paragraphs of that Act, which refer to the RTS on SCA.

In order to maintain equal competition between all stakeholders, the EBA seeks to maintain neutrality and defines basic rules of functioning in PSD2 at a general level. Despite the specification of open banking rules to third parties, the implementation across particular banks may vary. This versatility implies a more complex, and more expensive integration of third parties to become payment service providers keeping accounts. In order to support the European PSD2 directive and its proper implementation, the banks in the Czech market have decided to standardize their solutions.

The aim of the Czech Standard for Open Banking is to lay down rules for communication, mainly for services defined by the PSD2: Account Information Service Provider (AISP), Payment Initiation Service Provider (PISP), Card Issuing Service Provider (CISP)

The standard is designed and developed to maintain a high degree of versatility. The standard is voluntary, and it is up to each bank to consider joining the standard. With respect to the fact that systems and their operations differ across all payment service providers, standard participants may deviate from the standard at certain points, within the scope of their specific information systems. Since each payment service provider is obliged to document its solution according to PSD2, a large number of fields for optional information is defined in the Czech Open Banking Standard.

The main benefits of the Czech Standard are the easier integration of TPP into systems which banks are offering, including without any other intermediaries, ensuring a uniform interpretation of the PSD2 in the Czech market, compliance with the content of the transmitted data and security elements in communication with banks, support for the unified functioning of services for clients across banks, timely readiness for the implementation of PSD2.

The Czech Standard for Open Banking consists of these parts:

- Introduction and common features
 - Technical description

- Security standard
- Definition of API Account Information
- Definition of API Payment Initiation
- Definition of API Balance Check
- Examples of requests and response of particular API resources in JSON format

1 COBS Content

1.1 A General Description of Standard Content

This standard includes a definition of the individual areas of the interface for third party access to bank information and services. In terms of the definition of this interface in the required detail and with a certain degree of freedom, the individual descriptions break down into the specification of the technical interface, the security interfaces / principles and the data content of the individual services.

The technical description is intended to specify the basic parameters of the communication and justifies their use.

The Security standard primarily defines the principle of obtaining a user access authorization (bank client) to individual bank services for a third party.

Data content specifies an overview of the elements of individual services and the principles of their use.

1.2 Technical Description

1.2.1 Transport protocol

The HTTP 1.1 protocol is used as the transport communication protocol for this API according to [RFC7230, RFC7231, RFC7232, RFC7233, RFC7234, RFC7235] or the HTTP 2.0 protocol according to [RFC7540].

1.2.2 Interface architecture

The **REST** (Representational State Transfer) is used to design and implement the API communication interface. The use of hypermedia links is optional and is not part of the API specification (suggested interfaces using hypermedia links do not work).

1.2.3 Data entry format

JSON (JavaScript Object Notation) is used as a format for the (request) data entry and (response) API. Undefined or unused elements are returned from the API with a null value, and thus they are not hidden. Should it be required to define an element as blank or inaccessible, it is possible to return the element with

an empty string value. For blank fields, it is required to return an empty collection ([]) to the place of the null value.

1.2.4 Type of transmission media

Due to the use of the JSON data transfer format, this transmission character is defined by the relevant MIME/Content-type. For this API, the use of **application/json** as the content type is required, except for the said exceptions. The default coding of transfer content is UTF-8.

An example of the parameter in the request header:

```
Content-Type: application/json
```

1.2.5 Coding

The coding of the transfer character set is UTF-8 by default. A change in the desired coding of the character set can be done with the API, which enables this by using the HTTP request header **Content-Type** with the parameter **charset**.

An example of the change of coding for the parameter Content-Type:

```
Content-Type: application/json; charset=UTF-16
```

All URLs and query parameters need to conform to the syntax specified by the W3 Uniform Resource Identifier specification. In effect, this means that URLs must contain only a special subset of ASCII characters. And also alphanumeric symbols, and some reserved characters for use as control characters within URLs.

Below the table summarizes these characters:

SET	CHARACTERS	URL USAGE
Alphanumeric	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	Text strings, scheme usage (http), port (e.g 8080), etc.
Unreserved	- _ . ~	Text strings

Reserved	!*'();:@&=+\$,/?%#[]	Control characters and/or Text Strings
----------	----------------------	--

1.2.6 Localization

Unless otherwise defined for specific resources, it is possible to use, for the setting of response localization, the parameters of request and response headers **Accept-Language** a **Content-Language**.

Accept-Language is a request parameter and allow defining a set of preferred localizations arranged according to preferences.

Content-Language is a parameter of the response header and defines the localization used in the response to a request.

An example of the request with preferred localization:

```
Accept-Language: en,en-US,fr;q=0.6
```

An example of the response for the required localization:

```
Content-Language: en
```

1.2.7 HTTP methods used in API and their specifications

METHOD	INDEMPOTENT	CHANGES STATUS	PURPOSE
GET	Yes	No	Select: read-only calling (e.g., a list of client accounts)
POST	No	Yes	Creatae: Creates a new facility / event (e.g. creating a new transaction), calling the function

PUT	Yes	Yes	Update: Change of the existing facility (e.g. change of the transaction parameter)
DELETE	Yes	Yes	Delete: Deleting the existing facility (e.g. deleting a message)

GET, PUT and DELETE calls are idempotent. It does not matter what the server status is like, if the call is used only once or more (with the same parameters / content). POST calling, which usually creates new records may create duplicate data on a recalling.

1.2.8 Use of filtering, sorting and pagination of API elements

This section specifies the principles of filtering, sorting, and pagination of API content. These principles apply only to selected API resources. Each API resource in its description specifies whether any of these properties is available within resource usage.

In the case that the use of pagination or parameter filtering results in an empty element set, this empty set is also passed as a valid API call response (e.g., in comparison with the response status 404).

1.2.8.1 Filtering (fromDate, toDate, ...)

Specific parameters are defined for filtering for each API resource that allows filtering. These parameters define the range or scope of filtering and are explicitly defined for each such resource.

For example, for filtering a specific time interval, the query parameters **fromDate** and **toDate** can be introduced for the API resource.

Example of using parameters for filtering:

```
GET /transactions?fromDate=2016-05-12T05:37:30+02:00&toDate=2016-05-13T10:00:30+02:00
```

```
// It returns the list of transactions for the period defined from date and time stated in the parameter fromDate up to the date and time stated in the parameter toDate inclusive
```

1.2.8.2 Sorting on the part of API server (sort, order)

Specified API resources allow requesting data in a sorted order. The sorting property is explicitly listed in the description of each API resource. The API-specified sorting allows selecting an ascending or descending process for several fields at once with the option of defining the sorting direction for each field separately.

At the same time, the mechanism allows to define the ranking order by individual fields.

The following parameters are introduced for sorting

PARAMETER	MANDATORY	DEFAULT VALUE	PURPOSE
sort	Yes	-	It defines a list of fields for sorting. The individual fields in the list are separated by a comma
order	No	asc	It defines the sorting direction for each field listed in the sort parameter. Possible values are: <ul style="list-style-type: none">- asc for ascending sorting- desc for descending sorting The individual sorting directions are separated by commas precisely as the field names in the sort parameter and are listed in the same order as the fields listed in the sort parameter.

An example of a multiple-order sorting request where the type field does not have the specified direction and will be sorted by default in ascending order:

```
GET /accounts?sort=createDate,type,accountNumber&order=desc,,desc
```

1.2.8.3 Selection of specific fields (fields)

The API resources for which this is explicitly specified allow requesting only the specific elements in the request response. Such a request applies if the client does not require all fields but only the selected elements from API.

The required elements can be specified as a comma-separated list into the parameter **fields**.

An example, when the client requires from API in the response to return only the defined element (accountNumber, createDate, and type):

```
GET /accounts?fields=accountNumber,createDate,type
```

1.2.8.4 Pagination (page, size)

For specific API resources that return sets (such as a transaction field), it is possible to require a paged list. For this query, query parameters **page** and **size** are used. Each resource that allows requiring a paged list has this feature specified explicitly.

Parameters of query for page

PARAMETER	MANDATORY	DEFAULT VALUE	PURPOSE
page	No	0	It defines the required page number. Pages are numbered from 0. If the parameter is not specified, the API returns the first (zero) page.
size	No	If not specified, API will return the entire set	It defines the required number of records on the page. If the parameter is not specified, the API returns the entire set.

Resources that allow paging contain the specific structure in which the set of entries is presented. The structure always contains information about the data page such as the number of the next page, the number of pages, the number of entries on the current page, and possibly the total number of entries across all pages (optional data).

Parameters of response pagination

PARAMETER	MANDATORY	PURPOSE
pageNumber	Yes	Number of current page
pageCount	Yes	Total number of query pages
nextPage	No	The number of the next page. If the current page is at the same time the last page of the list, then this parameter is not listed or is given with a null value.
pageSize	Yes	Number of entries per page. This parameter can match the required size value from the query, except for the cases when it is the last page,

		or when the requested page range exceeded the maximum limit defined for the particular API resource.
totalCount	No	Optional data about the total number of entries for all pages. If this value cannot be precisely determined, it is not stated.
page/items/...	Yes	It is an element of field type that contains a set of required entries. The name of this element should match the contents of the set.

An example of query on paged API:

```
GET /transactions?size=25&page=1
```

This query returns the second page of records from the 26th record and further if the second page exists, or if there are more than 25 records.

An example of paged response:

```
{
  "pageNumber": 1,
  "pageCount": 12,
  "nextPage": 2,
  "pageSize": 25,
  "totalCount": 298,
  "page": [
    { ...item... },
    { ...item... },
    { ...item... }
  ]
}
```

1.2.9 Versioning of API

Should it be necessary to issue a new version of the API at the same time as the existing one, this information is part of the URI for the individual API resources. For each API, the API version is stated as a simple number from the sequence of versions, each time with the prefix “v”.

An example of version provided in URI of API resource:

```
GET /api/v1/
```

The API definition in this specification does not include versioning within defined URI resources.

1.2.10 HTTP status codes used and addressing errors in API

To communicate and manage exceptions, these APIs use uniformly established HTTP status codes and defined error content.

For each API and its resources, a list of possible error statuses with their relationship to the appropriate HTTP status code, is given.

List of HTTP status codes used in API

STATUS CODE	MEANING	PURPOSE
200	OK	Correct response with content
201	CREATED	Correct response = new entry created
204	NO CONTENT	Correct response = entry was deleted
304	NOT MODIFIED	Resource without change = possible to use cache data
400	BAD REQUEST	This is an invalid query that cannot be answered. E.g., if JSON content is not valid for this resource.
401	UNAUTHORIZED	Querying requires user authentication
403	FORBIDDEN	Access to the requested resource is not granted or is not possible for that user.
404	NOT FOUND	The requested object / page does not exist or was not found

415	UNSUPPORTED MEDIA TYPE	Request contains a request for unsupported transmission type (e.g. in relation to headers Accept and Accept-Language)
422	UNPROCESSABLE ENTITY	This error can be used if the requested object cannot be processed, or the required query parameter is absent.
500	INTERNAL SERVER ERROR	Server error that may be triggered by technical problems or in case of an untreated error status.
501	NOT IMPLEMENTED	It can be used if the server does not support the required operation.
503	SERVICE UNAVAILABLE	Server error triggered in case of temporary overload or scheduled maintenance.

1.2.10.1 Defined error content

For all error statuses addressed, specific error content is returned in JSON format. The content is defined by mandatory and optional parameters that the API client side can evaluate and use to properly route the response of the application or operation.

Since multiple error statuses can occur at the same time (e.g., multiple invalid elements when initiating a payment), the error always contains a set of errors defined by individual elements.

Parameters of error response

PARAMETER	TYPE	MANDATORY	PURPOSE
errors	Array	Yes	The superior element contains a set of all error statuses
errors.error	Text	Yes	It contains a specific error code
errors.parameters	Object	No	The field of additional elements specific for the given error code. These parameters are always specified in the description of a specific error code.
errors.scope	Text	No	It determines the JSON path of the request element that caused the error status.
errors.message	Text	No	Optional text description. It is not intended for interpretation to the end user but, for example, to enrich the error log.

An example of response with the set of errors:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{
  "errors": [
    {
      "error": "SOME_ERROR_CODE",
      "parameters": null,
      "message": null
    }, {
      "error": "ANOTHER_ERROR_CODE",
      "scope": "account.amount.currency",
      "parameters": null,
      "message": null
    }, {
      "error": "OTHER_ERROR_CODE",
      "parameters": {
        "AMOUNT_ENTERED": 10000,
        "CURRENCY": "EUR",
        "LIMIT": 500,
      }
      "scope": "orders[3].amount.value",
      "message": "Requested amount is too large"
    }
  ],
}
```

The error response status corresponds to the status defined for specific error codes or, for each error code defined in the API resource documentation, the HTTP response status that this error causes is also stated.

1.2.11 Authentication and Request Authorization (OAuth2)

API defined by this specification is not responsible for verifying user credentials (that the person for whom API is accessed is the one for which it impersonates, e.g. by checking the user's name and password), but is responsible for verifying that the person for whom the API is accessed has the right to obtain such data.

Therefore, the request authorization is based on the OAuth2 authorization flow concept secured with a token, and only has to check the validity of the tokens that the client provides for each call as evidence that it can access the requested data.

Within these APIs, the authorization token is considered to be a short-term and non-constitutive element that must be used in every API call that requires the request authorization.

The API provider interface uses a token to verify the client's communication privileges on a particular resource. Verification of the token is performed with an API service mechanism/provider and only upon this verification, the business logic linked to the required data source, is executed.

This API interface may work with any type of token if the following assumptions are valid:

- Token has the format / coding that may be transferred using HTTP of the field header "Authorization".
- Token the length of maximum 1024 bytes.
- It is possible (with the API Provider service) to verify the validity of the token.
- It is possible (with the API Provider service) to derive the user from the token.
- It is possible (with the API Provider service) to derive the authorization of the user from the token. This includes a separate authorization to read and write up to the level of individual business services/objects (e.g., one particular account, message). However, the Token should alternatively refer to privileges within a larger range of API Provider services, in particular entities owned by the user, e.g., "All own accounts".

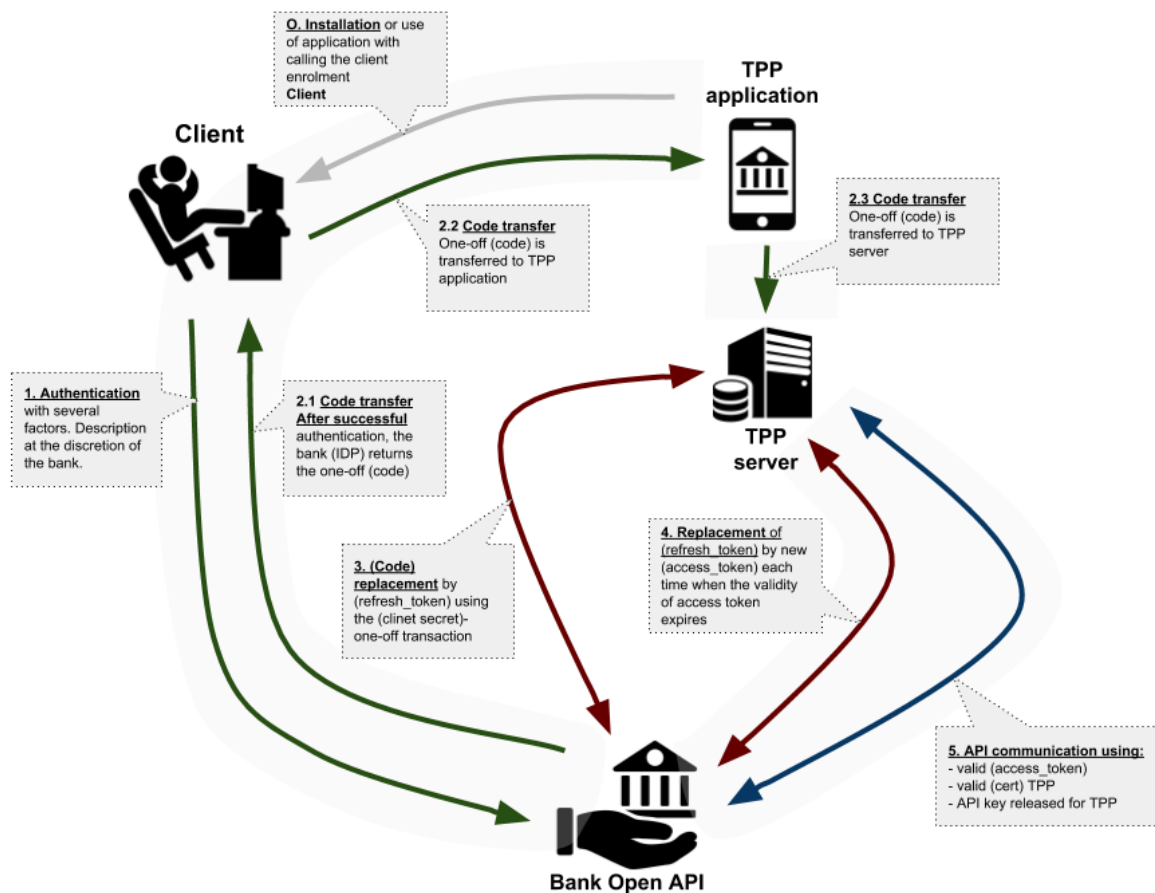
An example of using the token in the request header:

```
Content-Type: application/json
Authorization: Bearer dd3c4d08416c39af411ac0a37d110bea365fc1e
```

1.2.12 Security standard

The description of the client enrolment principle or of client product(s) at the bank to a third-party application. This principle is considered within this Standard. The described flow begins at the point where the client already has and in some way uses a third-party application.

1.3 Flow in the Process of Client Enrolment to the TPP Application



Flow enrolment chart

1.3.1 Individual flow steps

0. Installation - The client installs or accesses a third-party application and starts to use it. The Bank does not enter this process. During the use of the application, a third party allows the user, as a

client of the Bank, to implement his own product to the application that the client has at the Bank. The introduction of a third-party product and service corresponds to the role under which the third party is licensed (AISP, PISP, CISP). The application starts the client authentication process by redirecting it to the Bank's authentication point (the identity provider).

1. **Authentication** – The client, through the authentication process ensured by the Bank, goes through the authentication in the SCA principle. The authentication may be, for instance, federated and is in the regime of each bank with a character similar to the one known to the client, e.g. from Internet banking.
2. **Code transfer** – The result of successful client authentication is a generated one-time *code* which is transferred to the third-party application in the response when routed from par. 0., see 2.1 - 2.3.
3. **Refresh_token acquisition** - One-time replacement of the *code* by the *refresh_token*. For the replacement, the third-party must know the *client_secret* as a joint “secret” between the third party and the Bank (IDP of the bank). The *refresh_token* is valid for the relevant client and the specific third party within their role.
4. **Access_token acquisition** – Since to communicate with API, it is necessary to have a valid *access_token* for its acquisition it is necessary to execute the replacement of *refresh_token* by a valid *access_token* with a relatively short expiry period (e.g., 3 600s).
5. **Communication with API** - The API communication with the valid *access_token* and third-party certificate. Since the *access_token* is issued in the context of a specific user, and the third-party application is the request authorization at API executed with respect to the contract. At the same time, the validity of the third-party certificate and its role is verified.

The solution is the inspiration of providing authentication as a service with client impersonification in the sense of OpenID - <http://openid.net/> standard. It is an open standard enabling the decentralization of user authentication. In most cases, it is used as a substitute for the user authentication solution, with authentication and, in some cases, authorization executed by the operator of the OpenID service.

To generalize the solution, the basis for this proposal is only to use the OAuth2 open protocol to issue authorization tokens. Then primarily, the use of **code grant** authorization framework. See Chapter 1.4.1.1 OAuth2 Code Grant.

TERM/ACRONYM	DESCRIPTION
TPP	<i>Third party provider (AISP, PISP, CISP)</i>
SCA	<i>Strong Customer Authentication</i>
code	<i>One-time token/code usable only for a specific client and TPP in replacement by a long-term refresh_token</i>

refresh_token	<i>A long-term token issued based on replacement by the one-time code.</i>
access_token	<i>A short-term (in some cases, one-time) token that may be generated again using a refresh_token. The token serves for the authorization of request at API.</i>
IDP	<i>Identity provider</i>
STS	<i>Secure Token Service</i>

1.3.1.1 OAuth2 Code Grant

Within the OAuth2 protocol, in the case of **code grant** authorization framework, it is a method how `refresh token` and `access token` may be released to the partner application as a result of the user identification and authentication.

The partner application uses the short-term `access token` to communicate with the API of the bank and after its expiry, the `refresh token` may be used to demand a new `access token`.

1.3.1.1.1 Basic properties

- The `access token` is issued as short-term (e.g., 3 600s) and may be revoked (by the user, application and the provider/Bank)
- The `access token` is issued for a specific application and a specific user it may not be used successfully for another application
- It is not possible to use the `refresh token` directly for communication with API, it has a long or unlimited validity (e.g., in the case of PSD2, 90 days) and the provider may have the possibility to revoke it and the provider may allow the possibility to the user
- The provider (the Bank) and the application (TPP) share a joint “secret” - `client secret`
- The result of the user identification and authentication is a code that may be replaced by the third-party application by the `refresh token` and `access token` using the `client secret`
- It is not possible to use the `code` without knowing the `client secret`

1.3.1.1.2 Description of code grant flow

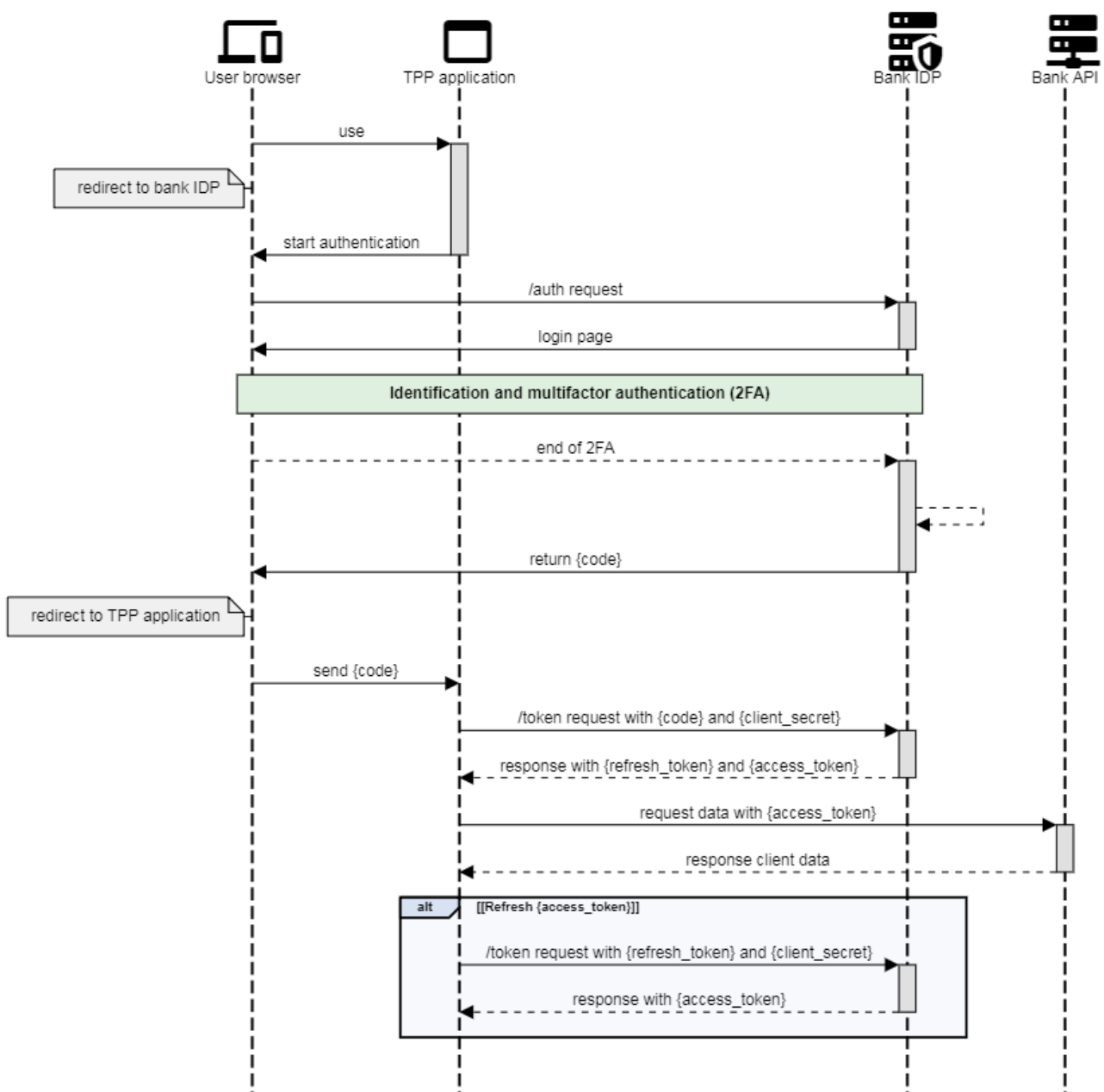
Flow initiation:

- The TPP application has its own unique `client_id` assigned from the provider and knows `_id` and `client secret` for the user
- On the issue of `client_id` and `client_secret`, the provider obtains the information on `redirect uri` - i.e. on URL where the user should be directed after successful authentication

Individual steps of code grant flow:

1. The TPP application directs the user to `/auth` resource of the provider (bank) to execute the identification and authentication

2. The client identification and authentication run - the steps are fully at the discretion of the provider
3. After successful authentication, the provider issues a `code`, and directs the client to the application URI (`redirect_uri`) with it
4. The TPP application will use the resource `/token` to obtain the `refresh_token` and the `access_token`. When calling the resource, the `code` and `client_secret` are delivered to the provider
5. The TPP application uses the `access_token` to communicate with the provider's API
6. The Provider will verify the `access_token` internally, e.g. using the internal STS (secure token service). On the verification, they acquire the user identity based on authentication of which the `access_token` was issued



1.3.1.1.3 Solution preconditions

The Bank ensures

- Issuing the resource for dynamic registration
- Issued authorization resource (**resource 1.**)
- Issued resource for the issue of refresh_token and access_token (**resources 2.a, 2.b**)
- Issued resource for the revocation (revoke) of access_token (**resource 3.**)
- Issuing the client_id and client_secret to a third party for its application
- For issued client_id keep the redirect_uri in the third-party application

TPP ensures

- Starting the authorization flow in bank URI
- Replacement of the code by refresh_token and access_token using the resource of the bank (**resource 2.a**)
- Replacement of the refresh_token by the access_token using the resource of the bank (**resource 2.b**)
- Secure keeping of client_secret, refresh_token, access_token and the code

1.4 An Overview of Resources to Ensure Security Flow Standard

1.4.1 Optional registration resource issued by the Bank

1.4.1.1 Recommended division of scopes into individual services within AIS and PIS

A recommended finer scope breakdown is described here. This division should allow third parties to request access to only the data and services they really need.

E.g. if as a TPP I only need a list of accounts and payment initiation, I can request the scopes aisp.account and pisp.payments. In this way, the end user does not consent to access to the transaction history and does not have to worry about an unnecessarily large amount of shared data. In turn, TPP knows what it can expect from ASPS.

ASPS must support at least the basic scopes of aisp, pisp, and cisp. The proposed breakdown is the recommended minimum, and ASPS can offer its own extended set of scopes for other supported services and data.

Scopes breakdown

SCOPE	DESCRIPTION	NOTE
aisp	Access to all services within AISP	This scope includes all others from AISP
aisp.accounts	Access to an overview of accounts	
aisp.balances	Access to information about balances	
aisp.transactions	Access to the overview of transactions	
aisp.directdebits	Access to the direct debits	
pisip	Access to all services within PIS	Does this scope include all others from PISP?
pisip.payments	Option to initiate payment	
pisip.directdebits	Ability to initiate a standing order	
pisip.accounts	Ability to obtain a list of accounts from those ASPs that offer a list of accounts together with PIS services	

Examples of use

TPP wants to initiate payments and keep track of accounts:

```
GET /oauth2/auth HTTP/1.1
Host: idp.banka.cz
Content-Type: application/x-www-form-urlencoded

client_id=MyPFM&
redirect_uri=https://www.mypfm.cz/start&
response_type=code&
scope=aisp.accounts pisp&
state=balance
```

TPP wants to initiate permanent payments and have an overview of accounts and transaction history:

```
GET /oauth2/auth HTTP/1.1
Host: idp.banka.cz
Content-Type: application/x-www-form-urlencoded

client_id=MyPFM&
redirect_uri=https://www.mypfm.cz/start&
response_type=code&
scope=aisp.accounts aisp.balance pisp.directdebit&
state=balance
```

1.4.1.2 0. Initialization / registration resource

By calling the resource, the TPP may demand the dynamic registration of client_id. It is necessary to use the valid certificate to call the resource. The output is the parameters client_id and client_secret that TPP needs to start and pass through the authentication process of the user (the client of the bank). And also the API_Key which is the carrier of the application configuration in calling the bank API.

Endpoint: POST <https://idp.banka.cz/oauth2/register>

Request content:

PARAMETER	VALUES	MANDATORY	DESCRIPTION
application_type	web, native	y	The type of application that will use the client_id. In the case of the web type, the definition of redirect_uri is required in the format of the web uri in the form http/s scheme. For the native type, in redirect_uri it is possible to enter, e.g. the application package, or the own format.
redirect_uri	Field containing the strings, e.g. in the URL or URN format [Max 3x 2047 B]	y	Enumeration of URL where the authentication flow is redirected at the end. The authentication request must contain just one of these registered URIs in the exact format.
authorization_redirect_uri	Field containing the strings, e.g. in the URL or URN format [Max 3x 2047 B]	n	Enumeration of URL where the authorization flow is redirected at the end. The authorization request must contain just one of these registered URIs in the exact format.
client_name	string [Max 255 B]	y	Name of the client application
client_name#en-US	string [Max 1024 B]	n	Name of the client application in the relevant language/coding.
logo_uri	URI	n	URI of the application logo (or the place from where it may be downloaded on the

	[Max 2047 B]		registration)
contact	string e-mail [Max 320 B]	n	E-mail as a contact to the responsible person on the part of the client application.
scopes	Field of strings [Max 10x 255 B]	n	The field of applications of required scopes. On the registration, scopes are validated against the content of the certificate used.

Response content (only new fields are stated against the request):

PARAMETER	VALUES	MANDATORY	DESCRIPTION
client_id	string	y	The client_id assigned by the application. This ID starts the authentication process and communication is decoded in the replacement of code and refresh_token.
client_secret	string	y	Client secret - password/token issued by the bank IDP for the TPP application (client_id)
client_secret_expires_at	Time	n	Default value is 0 (client_id never expires). Otherwise, the value is stated in seconds from data 1970-01-01T0:0:0Z
api_key	string	y	API key that the application uses to communicate with the bank API. If the bank API does not support API keys, the value „NOT_PROVIDED“ will be returned.

Error codes

HTTP STATUS	CODE	DESCRIPTION
400	INVALID_REQUEST	Invalid request. A mandatory field is missing in the request, or it is in an unsuitable / invalid format.
401	UNAUTHORIZED_CLIENT	The client is not authorized to execute the request.
401	ACCESS_DENIED	Authorization server denied an access.
500, 503	SERVER_ERROR	Authorization server error.
400	INVALID_SCOPE	Invalid request scope.

403	INSUFFICIENT_SCOPE	For instance, and insufficient authorization to use the required scope.
400	INVALID_REDIRECT_URI	The value of one or more redirect uri is not valid.
400	INVALID_AUTHORIZATION_REDIRECT_URI	The value of one or more redirect uri is not valid.

A request example:

```

POST /oauth2/register HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: idp.banka.cz

{
  "application_type": "web",
  "redirect_uris":
    ["https://www.mymultibank.cz/start",
     "https://www.mymultibank.cz/start2"],
  "authorization_redirect_uris":
    ["https://www.mymultibank.cz/init",
     "https://www.mymultibank.cz/init2"],
  "client_name": "Moje univerzální banka",
  "client_name#en-US": "My cool bank",
  "logo_uri": "https://www.mybank.cz/logo.png",
  "contact": "info@mybank.cz",
  "scopes": ["aisp", "pisp"]
}

```

A response example:

```
HTTP/1.1 201 Created
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{
  "client_id": "a0b25291f0",
  "client_secret":
    "AAjkk45sd78ad454gddd8712_4555g5g5g5gg",
  "client_secret_expires_at": 0,
  "api_key":
    "00000000-1212-0f0f-a0a0-123456789abc",
  "application_type": "web",
  "redirect_uris":
    ["https://www.mymultibank.cz/start",
     "https://www.mymultibank.cz/start2"],
  "authorization_redirect_uris":
    ["https://www.mymultibank.cz/init",
     "https://www.mymultibank.cz/init2"],
  "client_name": "Moje univerzální banka",
  "client_name#en-US": "My cool bank",
  "logo_uri": "https://www.mybank.cz/logo.png",
  "contact": "info@mybank.cz",
  "scopes": ["aisp", "pisp"]
}
```

1.4.1.3 0.1 Information on application registration data

By calling this resource, the TPP can request an overview of application-specific registration data. To call a resource, a valid certificate and client_id that is issued to this TPP, must be used. The output is an overview of the registration data

Endpoint: GET https://idp.banka.cz/oauth2/register/{client_id}

Response content:

PARAMETER	VALUES	MANDATORY	DESCRIPTION
client_id	string	y	The client_id assigned by the application. This ID starts the authentication process and communication is decoded in the replacement of code and refresh_token.
client_secret	string	y	Client secret - password/token issued by the bank IDP for the TPP application (client_id)
client_secret_expires_at	Time	n	Default value is 0 (client_id never expires). Otherwise, the value is stated in seconds from data 1970-01-01T0:0:0Z
api_key	string	y	API key that the application uses to communicate with the bank API. If the bank does not support API keys, the value is returned to the value „NOT_PROVIDED“
application_type	web, native	y	Type of application that will use the client_id. In the case of the web type, the definition of redirect_uri is required in the format of the web uri in the form http/s scheme. For the native type, in redirect_uri it is possible to enter, e.g. the application package, or the own format.
redirect_uri	Field containing the strings, e.g. in the URL or URN format	y	Enumeration of URL where the authentication flow is redirected at the end. The authentication request must contain just one of these registered URIs in the exact format.
authorization_redirect_uri	Field containing the strings, e.g. in the URL or URN format	n	Enumeration of URL where the authorization flow is redirected at the end. The authorization request must contain just one of these registered URIs in the exact format.
client_name	string	y	Name of the client application
client_name#en-US	string	n	Name of the client application in the relevant language/coding.

logo_uri	URI	n	URI of the application logo (or the place from where it may be downloaded on the registration)
contact	string e-mail	n	E-mail as a contact to the responsible person on the part of the client application.
scopes	String field	n	The field of applications of required scopes. On the registration, scopes are validated against the content of the certificate used.

Error codes

HTTP STATUS	CODE	DESCRIPTION
400	INVALID_REQUEST	Invalid request. A mandatory field is missing in the request, or it is in an unsuitable / invalid format.
401	INVALID_CLIENT	Invalid client_id.
401	UNAUTHORIZED_CLIENT	The client is not unauthorized to execute the request.
401	ACCESS_DENIED	Authorization server denied an access.
500, 503	SERVER_ERROR	Authorization server error.

A request example:

```
GET /oauth2/register/a0b25291f0 HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: idp.banka.cz
```

A response example:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```

{
  "client_id": "a0b25291f0",
  "client_secret":
    "AAjkk45sd78ad454gddd8712_4555g5g5g5gg",
  "client_secret_expires_at": 0,
  "api_key":
    "00000000-1212-0f0f-a0a0-123456789abc",
  "application_type": "web",
  "redirect_uris":
    ["https://www.mymultibank.cz/start",
     "https://www.mymultibank.cz/start2"],
  "authorization_redirect_uris":
    ["https://www.mymultibank.cz/init",
     "https://www.mymultibank.cz/init2"],
  "client_name": "Moje univerzální banka",
  "client_name#en-US": "My cool bank",
  "logo_uri": "https://www.mybank.cz/logo.png",
  "contact": "info@mybank.cz",
  "scopes": ["aisp", "pisp"]
}

```

1.4.1.4 0.2 Change of registration data

By calling this resource, the TPP can request a change of application-specific registration data. To call a resource, you need to use a valid certificate and `client_id` that is issued to this TPP. The output is an overview of the changed data

Endpoint: PUT https://idp.banka.cz/oauth2/register/{client_id}

Request content:

PARAMETER	VALUES	MANDATORY	DESCRIPTION
<code>application_type</code>	web, native	y	The type of application that will use the <code>client_id</code> . In the case of the web type, the definition of <code>redirect_uris</code> is required in the

			format of the web uri in the form http/s scheme. For the native type, in redirect_uris it is possible to enter, e.g. the application package, or the own format.
redirect_uris	Field containing the strings, e.g. in the URL format [Max 3x 2047 B]	y	Enumeration of URL where the authentication flow is redirected at the end. The authorization request must contain just one of these registered URIs in the exact format.
authorization_redirect_uris	Field containing the strings, e.g. in the URL or URN format [Max 3x 2047 B]	n	Enumeration of URL where the authorization flow is redirected at the end. The authorization request must contain just one of these registered URIs in the exact format.
client_name	string [Max 255 B]	y	Name of the client application
client_name#en-US	string [Max 1024 B]	n	Name of the client application in the relevant language/coding.
logo_uri	URI [Max 2047 B]	n	URI of the application logo (or the place from where it may be downloaded on the registration)
contact	string e-mail [Max 320 B]	n	E-mail as a contact to the responsible person on the part of the client application.
scopes	String field [Max 10x 255 B]	n	The field of applications of required scopes. On the registration, scopes are validated against the content of the certificate used.

Response content:

PARAMETER	VALUES	MANDATORY	DESCRIPTION
client_id	ID of TPP application	y	Unique identifier of the TPP application issued by the bank or the bank IDP. E.g., using the resource „0. Initialization/registration resource“
application_type	web, native	y	The type of application that will use the client_id. In the case of the web type, the definition of redirect_uris is required in the format of the web uri in the form http/s

			scheme. For the native type, in redirect_uris it is possible to enter, e.g. the application package, or the own format.
redirect_uris	Field containing the strings, e.g. in the URL format	y	Enumeration of URL where the authentication flow is redirected at the end. The authorization request must contain just one of these registered URIs in the exact format.
authorization_redirect_uris	Field containing the strings, e.g. in the URL or URN format	n	Enumeration of URL where the authorization flow is redirected at the end. The authorization request must contain just one of these registered URIs in the exact format.
client_name	string	y	Name of the client application
client_name#en-US	Any string	n	Name of the client application in the relevant language/coding.
logo_uri	URI	n	URI of the application logo (or the place from where it may be downloaded on the registration)
contact	string e-mail	n	E-mail as a contact to the responsible person on the part of the client application.
scopes	Field of strings	n	The field of applications of required scopes. On the registration, scopes are validated against the content of the certificate used.

Error codes

HTTP STATUS	CODE	DESCRIPTION
400	INVALID_REQUEST	Invalid request. A mandatory field is missing in the request, or it is in an unsuitable / invalid format.
401	INVALID_CLIENT	Invalid client_id.
401	UNAUTHORIZED_CLIENT	The client is not unauthorized to execute the request.
401	ACCESS_DENIED	Authorization server denied an access.
500, 503	SERVER_ERROR	Authorization server error.
400	INVALID_SCOPE	Invalid request scope.

403	INSUFFICIENT_SCOPE	For instance, and insufficient authorization to use the required scope.
400	INVALID_REDIRECT_URI	The value of one or more redirect uri is not valid.
400	INVALID_AUTHORIZATION_REDIRECT_URI	The value of one or more redirect uri is not valid.

A request example:

```

POST /oauth2/register/a0b25291f0 HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: idp.banka.cz

{
  "application_type": "web",
  "redirect_uris":
    ["https://www.mymultibank.cz/start",
     "https://www.mymultibank.cz/start2"],
  "authorization_redirect_uris":
    ["https://www.mymultibank.cz/init",
     "https://www.mymultibank.cz/init2"],
  "client_name": "Moje univerzální banka",
  "client_name#en-US": "My cool bank",
  "logo_uri": "https://www.mybank.cz/logo.png",
  "contact": "info@mybank.cz",
  "scopes": ["aisp", "pisp"]
}

```

A response example:

```
HTTP/1.1 200
Content-Type: application/json

{
  "client_id": "a0b25291f0",
  "client_secret_expires_at": 0,
  "application_type": "web",
  "redirect_uris":
    ["https://www.mymultibank.cz/start",
     "https://www.mymultibank.cz/start2"],
  "authorization_redirect_uris":
    ["https://www.mymultibank.cz/init",
     "https://www.mymultibank.cz/init2"],
  "client_name": "Moje univerzální banka",
  "client_name#en-US": "My cool bank",
  "logo_uri": "https://www.mybank.cz/logo.png",
  "contact": "info@mybank.cz",
  "scopes": ["aisp", "pisp"]
}
```

1.4.1.5 0.3 Deleting the application

By calling this resource, the TPP can request deleting the data and access to a specific application. To call a resource, a valid certificate and client_id that is issued to this TPP, must be used. The output is a confirmation of deletion.

Endpoint: DELETE https://idp.banka.cz/oauth2/register/{client_id}

Error codes

HTTP STATUS	CODE	DESCRIPTION
-------------	------	-------------

401	INVALID_CLIENT	Invalid client_id.
401	UNAUTHORIZED_CLIENT	The client is not unauthorized to execute the request.
401	ACCESS_DENIED	Authorization server denied an access.
500, 503	SERVER_ERROR	Authorization server error.

A request example:

```
DELETE /oauth2/register/a0b25291f0 HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: idp.banka.cz
```

A response example:

```
HTTP/1.1 201 Created
```

1.4.1.6 0.4 A request for new client_secret

By calling this resource, the TPP may demand the issue of a new client_secret. To call a resource, a valid certificate and client_id that is issued to this TPP, must be used. The original client_secret will be invalidated with the request.

Endpoint: POST https://idp.banka.cz/oauth2/register/{client_id}/renewSecret

A response content:

PARAMETER	VALUES	MANDATORY	DESCRIPTION
client_id	string	y	The client_id assigned by the application. This ID starts the authentication process and communication is decoded in the replacement of

			code and refresh_token.
client_secret	string	y	Client secret - password/token issued by the bank IDP for the TPP application (client_id)
client_secret_expires_at	Time	n	Default value is 0 (client_id never expires). Otherwise, the value is stated in seconds from data 1970-01-01T0:0:0Z

Error codes

HTTP STATUS	CODE	DESCRIPTION
400	INVALID_REQUEST	Invalid request. A mandatory field is missing in the request, or it is in an unsuitable / invalid format.
401	INVALID_CLIENT	Invalid client_id.
401	UNAUTHORIZED_CLIENT	The client is not authorized to execute the request.
401	ACCESS_DENIED	Authorization server denied an access.
500, 503	SERVER_ERROR	Authorization server error.

Example of request:

```
POST /oauth2/register/a0b25291f0/renewSecret HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: idp.banka.cz
```

Example of response:

```
HTTP/1.1 200 OK
```

```

{
  "client_id": "a0b25291f0",
  "client_secret": "BBjkk45sd78ad454gddd8712_4555g5g5g5gg",
  "client_secret_expires_at": 0
}

```

1.4.1.7 0.5 Request for a new API key – optional resource (if the Bank uses the API key)

By calling this resource, the TPP may demand the issue of a new API key. To call a resource, a valid certificate and client_id that is issued to this TPP, must be used. The original API key will be invalidated by the request.

Endpoint: POST https://idp.banka.cz/oauth2/register/{client_id}/renewKey

A response content:

PARAMETER	VALUES	MANDATORY	DESCRIPTION
client_id	string	y	The client_id assigned by the application. This ID starts the authentication process and communication is decoded in the replacement of code and refresh_token.
api_key	string	y	API key that the application uses to communicate with the bank API. If the API bank does not support API keys, the value „NOT_PROVIDED“ will be returned

Error codes:

HTTP STATUS	CODE	DESCRIPTION
400	INVALID_REQUEST	Invalid request. A mandatory field is missing in the request, or it is in an unsuitable / invalid format.
401	INVALID_CLIENT	Invalid client_id.
401	UNAUTHORIZED_CLIENT	The client is not unauthorized to execute the request.
401	ACCESS_DENIED	Authorization server denied an access.
500, 503	SERVER_ERROR	Authorization server error.

A request example:

```
POST /oauth2/register/a0b25291f0/renewKey HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: idp.banka.cz
```

A response example:

```
HTTP/1.1 200 OK

{
  "client_id": "a0b25291f0",
  "api_key": "00000001-1212-0f0f-a0a0-123456789abc"
}
```

1.4.2 Authentication resource issued by the Bank

1.4.3 1. Authentication resource

Endpoint: GET <https://idp.banka.cz/oauth2/auth>

A request content:

PARAMETER	VALUES	MANDATORY	DESCRIPTION
response_type	code	y	A mandatory parameter. It determines the authentication flow used. In this case, the code grant. For the authentication process it means that as the result of successful identification and authentication, a one-time code is expected instead of the

			access_token.
client_id	ID of TPP application	y	Unique identifier of the TPP application issued by the bank or the bank IDP. E.g., using the resource „0. Initialization/registration resource“
redirect_uri	URL	y	URL where the authentication flow is directed at the end. The URL is determined on the issue of client_id and within the authentication, the parameter is validated against the URL introduced for client_id in the IDP system of the bank. The value should be identical to one of the values introduced by using the resource „0. Initialization/registration resource“.
scope	List of authorizations separated by a space	n	It is a field of applications required by the scope (authorization). In the case of PSD2, it may be the roles of aisp and pisp. E.g., if the TPP is the holder of both authorizations, here it may ask for its application one of them or both, see a request example
state	Any string	n	This parameter may enrich the redirect_uri on the redirecting. It serves for the delivery of information from the application through the authentication flow.
login_hint	Any string	n	The parameter contains help to the authorization server regarding the user's login identifier. It can include a nickname, name, email, phone number, or anything that IDP uses to identify the user. The TPP can obtain this identifier from the end-user outside the authentication process.
id_token_hint	Any string	n	The id_token_hint parameter is used to pass a previously issued Token ID (if supported by this IDP) and serves as information about the current or past user authentication. If the user identified by the Token ID is already logged in or the login was performed using the Token ID, a positive response is returned from the IDP. If the ID Token received from the IDP is encrypted, the TPP must first decrypt the token. The TPP can re-encrypt the ID Token using the IDP public key if this allows.
request_uri	Any string	n	This parameter allows you to pass authorization elements by reference and not as individual parameters with a value. If this parameter is used, its value is a URL / string

			<p>representing a previously registered request object.</p> <p>The API for registering a request object is described in chapter 3.2.31, and in the context of this specification, is intended primarily to enable the initiation and authorization of payment with one SCA.</p> <p>A description of the response to an authorization request with a request object in request_uri can be found in chapter 3.2.32</p>
--	--	--	---

An example of URL for the authentication:

https://idp.banka.cz/**oauth2/auth**?state=profil&redirect_uri=https://www.mypfm.cz/start&client_id=MyPFM&response_type=balance&scope=aisp

A response content:

FIELD	DESCRIPTION
code	Authorization field
state	The state parameter from the TPP request

Error codes:

HTTP STATUS	CODE	REASON
302 400 see individual elements	invalid_request	<p>IDP returns this error if one of the mandatory parameters (e.g. nonce, state, ...) is missing from the request.</p> <p>It is also returned if the value of the parameter is invalid and the request cannot be executed except for exceptions specified by the own error code. See the following error codes.</p> <p>The error can also be used if an unsupported authentication or authorization method is requested (e.g. a POST request and the IDP only supports GET).</p> <p>In the case of status 400, this is an error status without a</p>

		redirect.
400	unauthorized_client	In the event that it is not possible to issue an authentication result for the given client and the grant variant used. For code_grant flow without session management, e.g. in the case of using an invalid client_id. Error status without redirect.
302	access_denied	Returned if authentication or authorization did not complete based on user action. Individual cases: <ul style="list-style-type: none"> • authentication incomplete by the end user • consent page rejected by the end user • authorization not completed by the end user
302	unsupported_response_type	It is returned if the requested response_type is not supported or is not registered in dynamic registration for the given client_id.
302	invalid_scope	It is returned in the event of a request for an unsupported scope or in the event that one of the requested scopes is not registered for the client_id used.
500	server_error	It is returned in case of an unexpected error on the server. Error status without redirect.
503	temporarily_unavailable	Returned if the server is temporarily too busy to return a response to the request. Error status without redirect.
302	interaction_required	It is returned if an authentication or authorization request was requested with the prompt=none parameter and it could not be completed without end-user interaction. It can be, for example, a request to enter a payment account for payment authorization, or another type of input. It is returned if it is a request for interaction other than login or consent confirmation.
302	login_required	It is returned if an authentication or authorization request was requested with the prompt=none parameter and it could

		<p>not be completed without end-user interaction.</p> <p>Such a case can occur, for example, when a user session on the IDP expires.</p>
302	account_selection_required	<p>It is returned if an authentication or authorization request was requested with the prompt=none parameter and it could not be completed without end-user interaction.</p> <p>Such a case can occur if the user has a valid session with the IDP, but multiple login accounts (e.g. with different scopes) and the end user is required to make a selection.</p>
302	consent_required	<p>It is returned if an authentication or authorization request was requested with the prompt=none parameter and it could not be completed without end-user interaction.</p> <p>Such a case can occur if there is a valid session, the user can be logged in, but confirmation of agreement with scopes is required.</p>
302	invalid_request_uri	<p>The error means that an invalid request_uri was used in the authentication or authorization request.</p> <ul style="list-style-type: none"> • request_uri is not registered in IDP • request_uri cannot be retrieved or its content is invalid
302	invalid_request_object	<p>The submitted request_uri is correctly registered and available, but contains an invalid value.</p>
302	request_uri_not_supported	<p>It occurs if an authentication or authorization request containing request_uri was sent to the server, but the server does not support this type of request.</p> <p>The IDP must state in its documentation that it does not support request_uri.</p>

A request example:

```
GET /oauth2/auth HTTP/1.1
Host: idp.banka.cz
```

```
Content-Type: application/x-www-form-urlencoded

client_id=MyPFM&
redirect_uri=https://www.mypfm.cz/start&
response_type=code&

scope=aisp pisp&
state=balance
```

A response example:

```
content-type: application/x-www-form-urlencoded
date: Wed, 8 Mar 2017 20:56:28 GMT
location: https://www.mypfm.cz/start?
         code=a200234062baa2ada828bbd33c1f6054&
         state=balance
status: 302
```

An example of error response:

```
HTTP/1.1 302 Found
Location: https://www.mymultibank.com/login?
         error=invalid_request
         &error_description=Unsupported%20response_uri
         &state=login_cz
```

Example request with login_hint:

```
GET /oauth2/auth HTTP/1.1
Host: idp.banka.cz
```


A request content:

FIELD	MANDATORY	DESCRIPTION
code	y	The authorization code returned from the authentication flow (code grant)
client_id	y	ID of TPP application
client_secret	y	Client secret - password/token issued by the bank IDP for the TPP application (client_id)
redirect_uri	y	Redirect URL identical to the URL delivered in the authentication request.
grant_type	y	According to the existing definition/practice, OAuth2 will be the value of authorization_code if the code is replaced by the refresh_token.

Error codes:

HTTP STATUS	CODE	DESCRIPTION
400	INVALID_REQUEST	Invalid request. A mandatory field is missing in the request, or it is in an unsuitable / invalid format.
401	UNAUTHORIZED_CLIENT	The client is not authorized to execute the request.
401	ACCESS_DENIED	Authorization server denied an access.
500, 503	SERVER_ERROR	Authorization server error.

A request example:

```
POST /oauth2/token HTTP/1.1
Host: idp.banka.cz
Content-Type: application/x-www-form-urlencoded

code=a200234062baa2ada828bbd33c1f6054&
client_id=MyPFM&
client_secret={client_secret}&
redirect_uri=https://www.mypfm.cz/start&
grant_type=authorization_code
```

A response example:

A successfully processed request will reply with a response with such a defined JSON payload:

```
{
  "expires_in": 3600,
  "token_type": "Bearer",
  "access_token": "ae9eef9b0af42c674d0b1c1128c37c2d"
  "refresh_token": "be9eef9b0af42c674d0b1c1128c37c2g",
  "acr": "0"
}
```

A response content:

FIELD	DESCRIPTION
access_token	A short-term (in some cases, a one-time) token that may be generated again using the refresh_token. The token serves for the authorization of the request at API.
refresh_token	A long-term token issued based on replacement by the one-time code.
expires_in	A time left before the expiration of access_token - in seconds.
token_type	A token type, e.g., "Bearer"
acr	[optional] Verification level. It may have the values between 0 and 4. Default 3. The value „0“ corresponds to nonSCA.

Note: Within the concept, it is possible to send other unspecified fields, in the response. The system should not assess the response as an error.

1.4.5 2.b Renewal of access token

The application may store the refresh token from the Get token resource and after the expiration of access_token, ask for a new one, through the refresh token. To that end, it is possible to use the Get token resource with these parameters:

Endpoint: POST https://idp.banka.cz/oauth2/token

A request content:

FIELD	MANDATORY	DESCRIPTION
client_id	n	ID of TPP application
grant_type	y	According to the existing definition/practice, OAuth2 will be the value of refresh_token, if the access_token is replaced by the refresh_token.
refresh_token	y	Valid refresh_token for which the replacement is carried out, e.g. be9eef9b0af42c674d0b1c1128c37c2g

Error codes

HTTP STATUS	CODE	DESCRIPTION
400	INVALID_REQUEST	Invalid request. A mandatory field is missing in the request, or it is in an unsuitable / invalid format.
401	INVALID_GRANT	Invalid authorization. E.g., an invalid refresh token.
401	UNAUTHORIZED_CLIENT	The client is not authorized to execute the request.
401	ACCESS_DENIED	Authorization server denied an access.
500, 503	SERVER_ERROR	Authorization server error.

A request example:

```
POST /oauth2/token HTTP/1.1
Host: idp.banka.cz
Content-Type: application/x-www-form-urlencoded

grant_type=refresh_token&
refresh_token=be9eef9b0af42c674d0b1c1128c37c2g
```

A response example:

A successfully processed request will reply with a response with such a defined JSON payload:

```
{
  "expires_in": 3600,
  "token_type": "Bearer",
  "access_token": "ae9eef9b0af42c674d0b1c1128c37c2d",
  "acr": "3"
}
```

A response content:

FIELD	DESCRIPTION
access_token	A short-term (in some cases, a one-time) token that may be generated again using the refresh_token. The token serves for the authorization of the request at API.
expires_in	A time left before the expiration of access_token – in seconds.
token_type	A token type, e.g., “Bearer”
acr	[optional] Verification level. It may have the values between 0 and 4. Default 3 or 4. The value „0” automatically corresponds to a nonSCA. The values 1 - 4 correspond to values defined by the ISO 29115 standard.

Note: Within the concept, it is possible to send other unspecified fields, in the response. The system should not assess the response as an error.

1.4.6 2.c Token introspection

Endpoint: POST <https://idp.banka.cz/oauth2/introspect>

Request parameters:

PARAMETER	DESCRIPTION
token	OAuth2 access or refresh token acquired based on the authentication process after the replacement by code or refresh token (in the case of access_token)
token_type_hint	Indicates the type of token being passed. Valid values are access_token , id_token and refresh_token .

Error codes

HTTP STATUS	CODE	DESCRIPTION
400	INVALID_REQUEST	Invalid request. A mandatory field is missing in the request, or it is in an unsuitable / invalid format.
401	UNAUTHORIZED_CLIENT	The client is not authorized to execute the request.
401	ACCESS_DENIED	Authorization server denied an access.
500, 503	SERVER_ERROR	Authorization server error.

Recommended validity of individual tokens:

TOKEN	VALIDITY [s]	NOTE
code	3600 (1h)	The design should be taken as the maximum value ensuring a reliable authentication process and exchange of tokens on the part of IDP and TPP.
access_token	300 (5m) - 3600 (1h)	Recommended scope. The 300s is offered as a solution to the RTS requirement for 5 minutes of inactivity.
refresh_token	7776000 (90d)	The proposed validity is based on a solution for a 90-day SCA exemption.
id_token	7776000 (90d)	

A request example:

```
POST /oauth2/introspect HTTP/1.1
Host: idp.banka.cz
Content-Type: application/x-www-form-urlencoded
```

```
token=be9eef9b0af42c674d0b1c1128c37c2g
&token_type_hint=access_token
```

Response parameters:

FIELD	DESCRIPTION
active	Indicates whether the token is active or not.
aud	The audience of the token.
client_id	The ID of the client associated with the token.
scope	A space-delimited list of scopes.
iss	The issuer of the token.
exp	The token expiration in seconds.
iat	The issuing time of the token in seconds since January 1, 1970 UTC.

A example response for valid non-expired token:

```
{
  "active": true,
  "client_id": "1b4005b4-233b-4693-ae9-4b7f39d74201",
  "scope": "AISP PISP",
  "aud": "https://protected.example.net/resource",
  "iss": "https://server.example.com/",
  "exp": 2500,
  "iat": 1419350238
}
```

A example response for expired:

```
{  
  "active": false  
}
```

1.4.7 3. Token invalidation

A function to invalidate an access or refresh token.

Endpoint: POST <https://idp.banka.cz/oauth2/revoke>

PARAMETER	DESCRIPTION
token	OAuth2 access or refresh token acquired based on the authentication process after the replacement by code or refresh token (in the case of access_token)

Error codes

HTTP STATUS	CODE	DESCRIPTION
400	INVALID_REQUEST	Invalid request. A mandatory field is missing in the request, or it is in an unsuitable / invalid format.
401	INVALID_GRANT	Invalid authorization. E.g., an invalid refresh token.
401	UNAUTHORIZED_CLIENT	The client is not unauthorized to execute the request.
401	ACCESS_DENIED	Authorization server denied an access.
500, 503	SERVER_ERROR	Authorization server error.

A request example:

```
POST /oauth2/revoke HTTP/1.1  
Host: idp.banka.cz  
Content-Type: application/x-www-form-urlencoded  
  
token=be9eef9b0af42c674d0b1c1128c37c2g
```

1.4.8 Error codes used in the enrolment

An overview and description of the error statutes used in the enrolment flow (TPP registration and authentication). The format and use are based on the OpenID Connect standard.

Error elements:

CODE	MANDATORY	DESCRIPTION
error	y	Contains the error code
error_description	n	Expanded text description of the error

Error codes:

HTTP STATUS	CODE	DESCRIPTION
400, 302	INVALID_REQUEST	Invalid request. A mandatory field is missing in the request, or it is in an unsuitable / invalid format.
401, 302	INVALID_CLIENT	Invalid client_id.
401	INVALID_GRANT	Invalid authorization. E.g., an invalid refresh token.
401	INVALID_TOKEN	An invalid token used.
401, 302	UNAUTHORIZED_CLIENT	The client is not authorized to execute the request.
302	ACCESS_DENIED	Authorization server denied an access.
500, 503	SERVER_ERROR	Authorization server error.
400	INVALID_SCOPE	Invalid request scope.
403	INSUFFICIENT_SCOPE	For instance, and insufficient authorization to use the required scope.
400	INVALID_REDIRECT_URI	The value of one or more redirect uri is not valid.

An example of error response:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json

{
  "error": "invalid_request",
  "error_description": "One or more values are invalid"
}
```

1.4.9 TPP authentication

The precondition of the solution is the use of a qualified eIDAS certificate to identify and authenticate a communicating third party. The directive allows the use of a web certificate QWAC (Qualified Website Authentication Certificate) or seal certificate QSealC (Qualified electronic Seal Certificate). Each certificate has a specific method of use.

The requirement to use the certificate is listed in this document as a specific flag for each of the defined requests. If a request to use the certificate is not specified in the request definition, the certificate is not required or the use of the certificate is purely at ASPSP (such as OAuth2 token requests).

In addition, the used certificate conforms to the eIDAS methodology should also meet the requirements of EBA-RTS. This means that it contains the data needed to successfully verify (authenticate) the TPP, its license, and a list of the services being treated. Best therefore, if it is an eIDAS certificate corresponding to standard ETSI TS 119 495.

1.4.9.1 Web certificate QWAC

Using this certificate would mean using a two-way (mutual) TLS (the Transport Layer Security protocol) as a successor of SSL (Secure Sockets Layer). The certificate would be used to establish TLS communication on both sides of the connection so that a qualified web certificate would have to be used by the bank (ASPSP) and TPP. The third party would be identified by validating the validity and content of its web certificate.

1.4.9.2 Stamps (electronic signature) QSealC

Using a seal certificate in this solution would mean requiring a request to be signed by a third party when communicating with the bank (ASPSP). The third party would be verified in this model by evaluating the validity of the signature of the message and the content of the public key sent along with the signature (for example, according to the CAdES standard). For proper use of QSealC, it is necessary to use signature on

clearly defined parts of requestu / response in a predefined structure. Because only so that the content of the signed request can be verified correctly on both sides (TPP and ASPSP). This definition is not part of this COBS version.

1.4.10 Communication security

Recommended methods of communication security when using a TLS (web certificate) for TPP authentication to communicate with the ASPSP API. The goal is not to duplicate existing TLS mechanisms but to reduce vulnerability despite the known weaknesses of these protocols.

1.5 Common http headers

List of by standard defined parameters in request and response headers.

For single requests and responses in this standard, only resource-specific header parameters are listed, ie those that are not listed among these general parameters.

1.5.1 Request headers

The following table contains the defined set of request header parameters common to most of the requests in this specification.

PARAMETER	TYPE	MANDATORY	PURPOSE
Content-Type	Max50Text	Yes	Defines MIME media type of the resource. For example application/json or application/x-www-form-urlencoded (OAuth2 /auth resources) and application/xml in case of bulk payment initiation.
Authorization	Max1024Text	Conditional (see comments for each resource)	Contains the credentials to authenticate a user agent with a server. Element is defined in RFC 6750 In the case of this specification as Bearer Token Usage.
API-key	Max100Text	No	The API key issued for TPP is part of the communication configuration for better identification of third-party call access settings. In essence, this is not a security element.
Accept	Max50Text	No	The Accept request HTTP header specifies which content types, expressed as MIME types, the client is able to understand.
Accept-Language	Max50Text	No	The Accept-Language request HTTP header specifies which languages the client is able to understand, and

			<p>which locale variant is preferred. If no `Accept-Language:` is given, local country language is assumed.</p> <p>For example:</p> <pre>Accept-Language: *</pre> <pre>Accept-Language: en-US, en;q=0.8</pre>
X-Request-ID	Max60Text	Yes	<p>Unique identifier for each request specified by TPP. It is recommended to use a UUID format with a length of 36 to 60 characters.</p>
Date	DateTime	Yes	<p>Each request of a transaction contains the date and time at which the message was originated. In timestamp format.</p>
Content-Length	Number	No	<p>The Content-Length entity-header field indicates the size of the entity-body, in decimal bytes, sent to the recipient.</p>
Transfer-Encoding	Text	Conditional	<p>The Transfer-Encoding general-header field specifies the form of encoding used to safely transfer the request payload to the ASPSP and back to TPP.</p>
Action-ID	Max60Text	No	<p>IDs that identify the technical or business process within the call of each API request.</p> <p>For example, it can be used to identify the process of retrieving a transaction history that is provided as a paged response available through multiple requests.</p>
User-Involved	Boolean	Yes	<p>The flag identifies if the end user is online and the request was made based on his activity.</p>
User-IP-Address	Max50Text	No	<p>Identifies the API's end-user IP address. The TPP should fill this value if the IP address can be detected. The address can be in IPv4 or IPv6 format.</p>
User-IP-Port	Max40Text	No	<p>An element used to pass on information about what end-user IP port is using.</p>
User-Device-OS	Max100Text	No	<p>The element is used to forward information about the end user operating system used.</p>

User-User-Agent	Max200Text	No	Contains information about the end-user web browser. The format should be the same as the standard HTTP parameter User-Agent.
User-Timestamp	DateTime	No	Local time on PSU device.
User-Geo-Location	Max100Text	No	End-user GPS coordinates (if the coordinates can be reliably detected). Required format: GEO:Latitude;Longitude Based on [RFC2426] in signed degrees format (DDD.dddd) example: GEO:50.089354;14.400031
User-Device-ID	Text	No	It contains a unique identifier of end user equipment, if available. For example, a unique mobile device identifier (IMEI).
TPP-Name	Text	Yes	The name of the original TPP that created the request. Eg. 'Star corporation, a.s.'.
TPP-Identification	Text	No	The identification (licence number) of the original TPP that created the request. Eg. 'CZ013574-15'.
Signature	Text	Conditional	If the TPP signs the request, it contains a digital signature.
Digest	Text	Conditional	When an TPP includes a signature he also must include a "Digest" header as defined in [RFC3230].
TPP-Signature-Certificate	Text	Conditional	The public TPP certificate used for signing the request, In base64 encoding (with the "BEGIN CERTIFICATE" and "END CERTIFICATE" markers).

1.5.2 Response headers

The following table contains the required set of response header parameters common to most of the requests/responses in this specification.

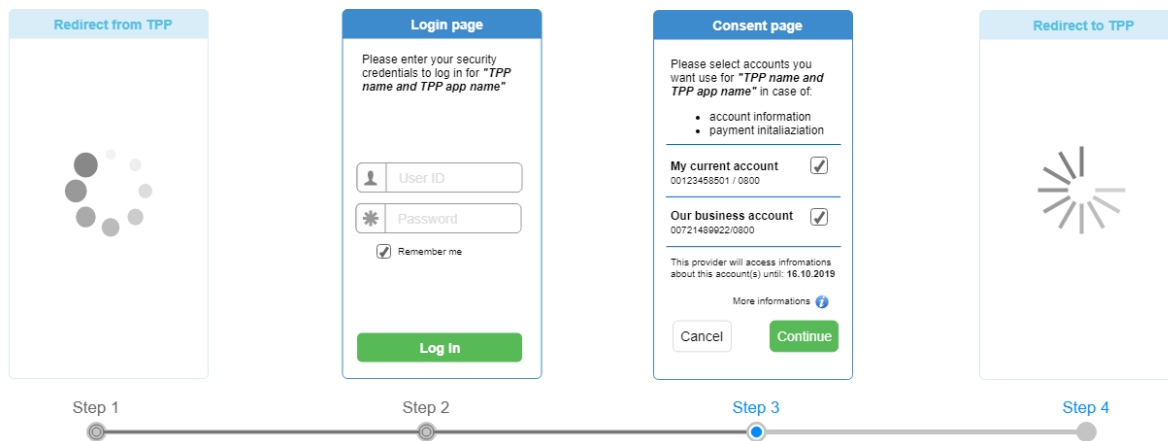
<i>PARAMETER</i>	<i>TYPE</i>	<i>MANDATORY</i>	<i>PURPOSE</i>
Content-Type	Max50Text	Yes	Defines MIME media type of the response. For example application/json or application/xml in case of bulk payment initiation.
X-Request-ID	Max60Text	Yes	Unique identifier for each request specified by TPP returned in response header.
Action-ID	Max60Text	No	IDs that identify the technical or business process within the call of each API request. APSPS should have a return value in this case.
Signature	Text	Conditional	If the ASPSP signs the response, it contains a digital signature.
Digest	Text	Conditional	When the ASPS in response includes a signature he also must include a "Digest" header as defined in [RFC3230].
ASPSP-Signature-Certificate	Text	Conditional	The public ASPSP certificate used for signing the response, In base64 encoding (with the "BEGIN CERTIFICATE" and "END CERTIFICATE" markers).

1.6 Consent Management

1.6.1 Consent flow

Recommended inclusion of the account selection page in the authentication flow of the bank. This is primarily an information page where the user (the client of the bank) has the option to select the accounts to be made available to the AIS or PIS services. The site also includes information on services and authentication validity.

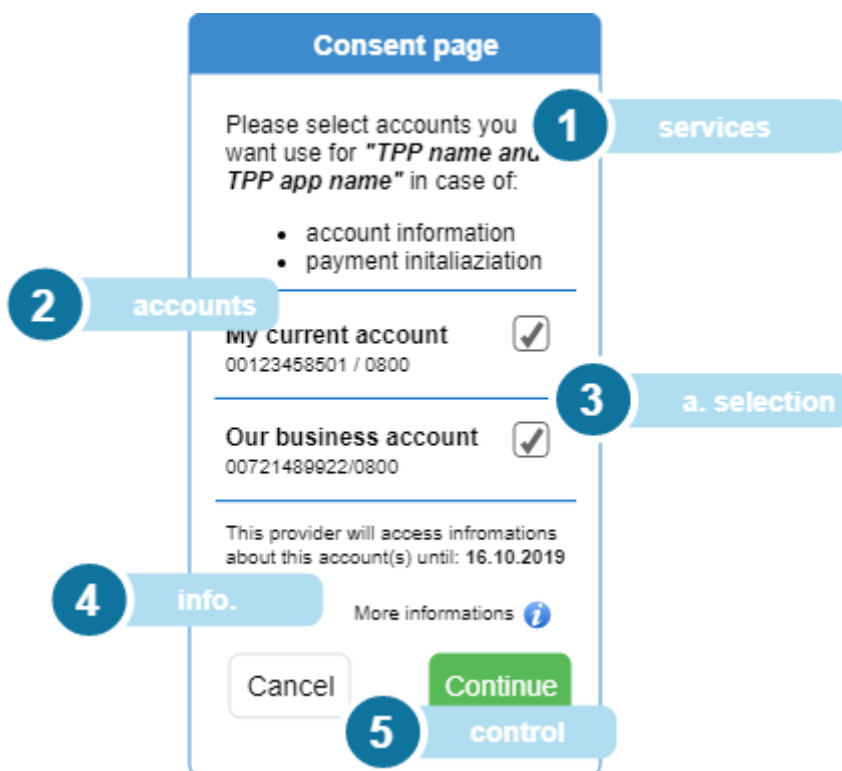
The goal is to standardize this page so that the ensure solution at different banks is similar and contains the same information.



Authentication flow

(1 – redirect from the TPP application 2 – SCA login process of the bank, 3 – account selection(s) and service information, 4 – redirect back to TPP application)

Account selection page and available product information:



Parts of the account selection and product information page

NUMBER	PART	DESCRIPTION	CONTENT
1	Services overview	Information about provided services	Text contains overview of informations for who and how is service provided.
2	List of accounts	List of PSU's payment accounts	The account list contains at least an identifier (account number / iban) and a user-friendly account name.
3	Account selection	An element allowing to "select" the account for which the service will be provided	A checkbox, switch, or other element that allows you to control and visualize your account / product choices.
4	Informations	Possibility to obtain additional information to the provided service. E.g. validity period, overview of provided data ...	At least a link to information about PSD2 services. E.g. pop-up window or drop-down list.
5	Page control	-	Resume / Cancel button (or others)

1.6.2 Get list of consents (GET /my/consents)

Resource to obtain a list of consents.

Resource characteristics

Mandatory: no

URI: /my/consents

HTTP Method: GET

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: no

Sorting: no

Filtration: no

Query parameters of the request: **not defined**

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is always mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of GET request for calling, please see Chapter 1.6.1.2 MESSAGE ELEMENTS Response to list of consents

Error codes defined for the service GET list of consents:

<i>HTTP STATUS CODE</i>	<i>ERROR CODE</i>	<i>PURPOSE</i>
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
401	UNAUTHORISED	Invalid/missing certificate = provider is not authenticated
403	FORBIDDEN	Calling of the method which does not correspond to the licence, or invalid certificate.

1.6.2.1 MESSAGE ELEMENTS Request to list of consents

This service has no content request.

1.6.2.2 MESSAGE ELEMENTS Response to list of consents

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+	consentId	[1..1]	Max45Text	Unique id of a specific consent for a particular PSU
+	consents	[1..1]	Array of products with consents	Collection of consents that the user grants to a particular TPP and its application.
++	identification	[1..1]	±	Payer account identifier
+++	Iban	[1..1]	IBAN2007Identifier	IBAN
+++	Other	[0..1]	Max35Text	Another payer account identifier, e.g. the account number.
++	accesses	[1..1]	Array of Access scopes	Array of scopes for specific product. Scopes are defined by the bank, including their granularity. From the point of view of PSD2, scopes can only be defined for individual PSD2 roles (such as AIS, PIS, PII) or, more gently, for individual services (balances, transactions, ...). This is in the hands of each bank.
++	validUntil	[0..1]	DATETIME	Date and time when the consent for this product expires.

1.6.3 Consent delete (DELETE /my/consents/{consentId})

Resource characteristics

Mandatory: no

URI: /my/consents/{consentId}

HTTP Method: GET

Authorization: the authorization of user/client as part of API calling **is optional**

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: no

Sorting: no

Filtration: no

Query parameters:

<i>PARAMETER</i>	<i>TYPE</i>	<i>MADATORY</i>	<i>PURPOSE</i>
consentId	Text	Yes	ID of the corresponding consent object as returned by an GET Consent Request

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is optional for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

Error codes defined for the service DELETE Consent

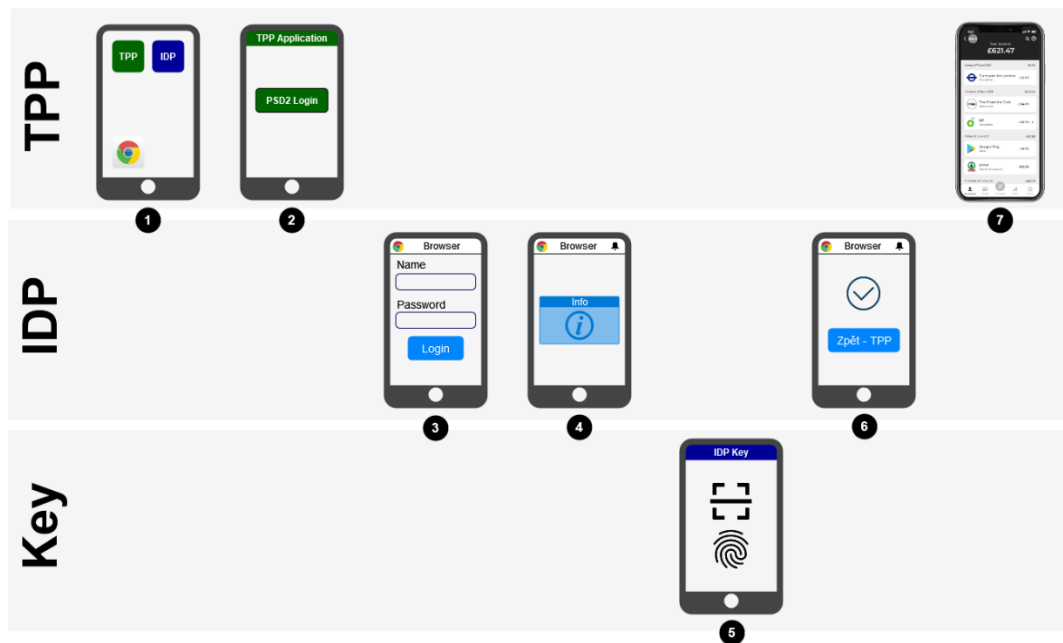
<i>HTTP STATUS CODE</i>	<i>ERROR CODE</i>	<i>PURPOSE</i>
401	UNAUTHORISED	Missing certificate.
403	FORBIDDEN	Calling of the method which does not correspond to the licence, or invalid certificate.
404	NOT_FOUND	Request for non-existing or invalid consent ID. Only if the correct Authorization attribute in the request header is used.

1.7 App2App communication

Applications with Open banking APIs, which are primarily intended for use on mobile devices, are increasingly being used. In such a case, it can be somewhat complicated to pass SCA (authentication) if ASPS uses a software token represented by a mobile application for authentication.

The OAuth2 and possibly OIDC authentication flow uses HTTP redirects for redirection to the selected authentication method and for the response of the authentication itself.

1.7.1 An example of an existing flow on a mobile device



1. The user has a mobile phone with the TPP application and his bank's Key
2. When using the TPP application, the user is prompted to authenticate to the account
3. You will be redirected to the ASPSP federated authentication page (IDP bank)
 - a. The user may have difficulty selecting the correct application to continue
4. After entering identification data (various methods, e.g. name and password), information will be displayed to continue in the Bank Key
 - a. The user may have a problem with receiving and confirming the notification
5. The user performs SCA verification in the bank key
 - a. From the key, the user must find the browser application himself (step 6)
6. In the meantime, the key usage information has been updated in the browser
7. The user finds the bookmark of the TPP application and continues to use it

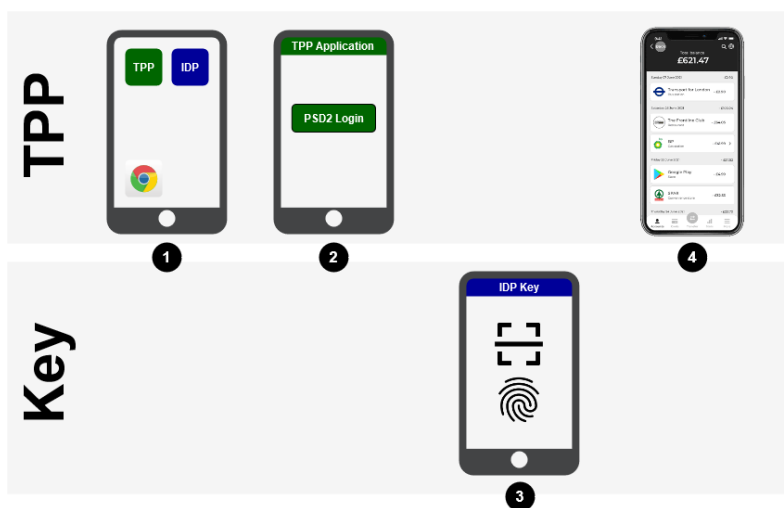
This flow can be too complex for the end user and leads to several places where an error can occur and, as a result, authentication cannot be completed. A variant of the solution is the use of already established Universal and App links, which are offered by individual mobile operating systems and are widely used.

Solution variant using Universal and App links

This solution assumes that both the TPP mobile application and the mobile authentication "key" (software token) have their auth URL (defined for IDP ASPS in chapter 1.4.3) and redirect_url as universal or app link (depending on the operating system) registered.

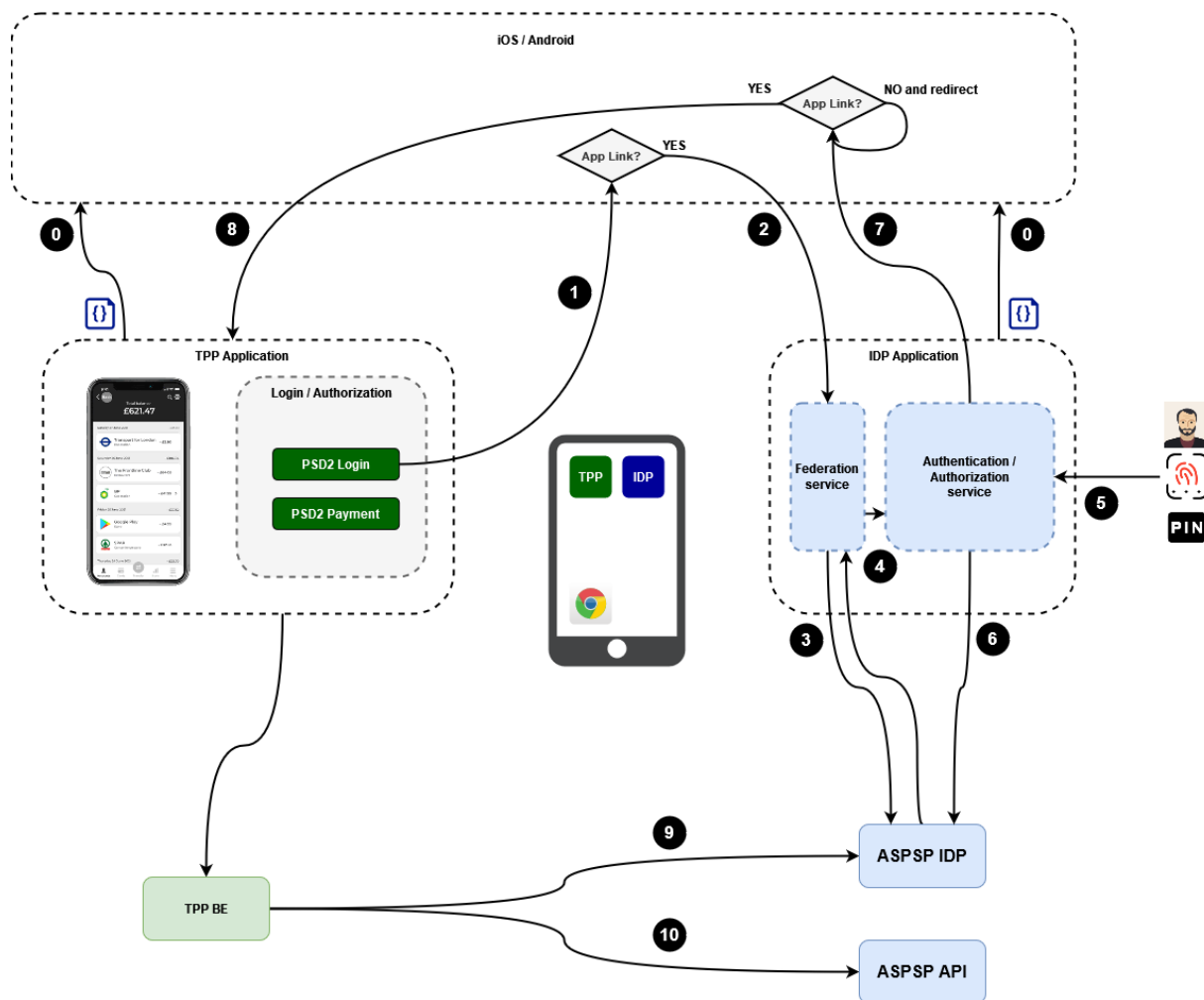
Thanks to this, the processing of auth and redirectet_url will be passed directly in the application and the launch of the Internet browser on the mobile device will be omitted.

1.7.2 Example flow using universal and app links



1. The user has a mobile phone with the TPP application and his bank's Key
2. When using the TPP application, the user is prompted to authenticate to the account
3. There will be a redirection to the ASPSP federated authentication page (IDP of the bank), which is, however, intercepted by the system as an app link of the bank's key, and the key will be launched, where the user will authenticate using SCA.
4. After finishing the SCA, the bank key redirects to the redirect_uri of the TPP application. This URI has TPP applications registered as an app link, so the application will be opened directly.

1.7.3 Description of interactions when using universal and app link flow



0. The TPP application and IDP Klíč have a manifest with App/Universal lines introduced in the system (Android/iOS).
1. In the TPP application, the user is redirected to the authentication endpoint of the IDP bank (the same one as for the federated authentication page)
2. According to the established manifest, the system transfers the processing of the request to the Bank Key application
3. The authentication application validates the request against the IDP (also transmits the user/device identification)
4. The IDP will transmit the same data as the "key wake-up"
5. The user performs SCA using one of the methods (biometrics, pin, password...)
6. The key will verify and authenticate the user (as it already does)
7. The key performs a redirect (intent) to the redirect_uri introduced with the TPP application.
8. Since the TPP application has this URL registered in the system through the manifest, the system will pass the processing of the URL to the TPP application.
9. TPP will exchange code from the redirect for tokens via the backend server
10. TPP consumes PSD2 bank API

1.8 Nomenclature and basic concepts identical across COBS

NOMENCLATURE / ACRONYM / TERM	DESCRIPTION
API	Application Programming Interface - defines a machine-accessible interface for application programming
HTTP	Hypertext Transfer Protocol – Internet protocol
JSON	JavaScript Object Notation - platform-independent way of writing data (data format)
OAuth2	Protocol for request authorization
Open banking services	General name for PSD2 services
REST	As the Representational State Transfer - interface architecture designed for distributed instruments
PIS service	Payment initiation
AIS service	Account information
CIS service	Balance Check
TPP with PIS service	The provider of payment initiation service
TPP with AIS service	The provider account information service
TPP with CIS service	Issuer of card payment instrument
TPP	Provider of the <service name> service

1.9 Health Check and Planned Outages generic resources

Generic resource defined as a possible extension of each API described in this specification. The bank issuing each API can use this resource to inform third parties about the availability of a particular service.

It is a universal REST JSON resource that returns information about the availability and status of each API. In addition to the basic status of the API, the bank can also inform about the individual parts of the service.

The use of this resource and its specific specification should be recorded in the resulting documentation of each bank in the enhanced API description.

1.9.1 Health Check

A description of the HealthCheck resource with an example for the definition of the AIS API.

Resource characteristics

Mandatory:	no
URI:	aisp/healthcheck
HTTP Method:	GET
Authorization:	no
Use certificate:	no
Paging:	no
Sorting:	no
Filtering:	no
Scopes:	no

Input character set: no restrictions in relation to chapter 3.1.1

Query parameters of the request:

PARAMETR	TYPE	MANDATORY	PURPOSE
-	-	-	-

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of GET response for calling, please see Chapter 1.8.1.1 RESPONSE MESSAGE ELEMENTS Helath Check service

1.9.1.1 RESPONSE MESSAGE ELEMENTS Health Check

LEVEL	MESSAGE ELEMENT	OCCURENCE	FORMAT TYPE	PRESENTATION
+	status	[1..1]	Text	Status of AISP API availability. See the possible statuses table in Chapter 1.8.1.2
+	description	[0..1]	Text	API description.
+	version	[0..1]	Text	Version of the API
+	time	[1..1]	Number	API status time
+	outage_planned_until	[0..1]	Text	If it is a planned outage, this is the date until which the outage lasts.
+	poll_interval	[0..1]	Text	Recommended interval that determines how often the resource calls for information. Frequent calls can lead to HTTP error 424.
+	details	[0..1]	±	Optional collection of API sub-services with their own states. As with the AIS API, this can be an account overview, balance information, and transaction overview. In this way, the bank can say that, for example, the API as a whole works up to a

				subset (such as an overview of account transactions).
++	accounts	[0..1]	±	Example for AIS accounts list service.
+++	status	[0..1]	Text	
+++	description	[0..1]	Text	
+++	time	[0..1]	Number	
++	balances	[0..1]	±	Example for AIS balance information service.
+++	status	[0..1]	Text	
+++	description	[0..1]	Text	
+++	time	[0..1]	Number	
++	transactions	[0..1]	±	Example for AIS transactions list service.
+++	status	[0..1]	Text	
+++	description	[0..1]	Text	
+++	time	[0..1]	Number	

1.9.1.2 Status list

STATUS NAME	DESCRIPTION
OK	The entire API or a specified portion (detail) of the API is available .
OUTAGE	The entire API or a specified portion (detail) of the API is temporary unavailable .
ERROR	The entire API or a specified portion (detail) of the API is unavailable .

1.9.2 Example of request and response for Health Check on AIS API

1.9.2.1 Example of request

Request headers:

```
GET https://api.bank.cz/aisp/healthcheck
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:21:01 GMT
API-key: 00000000-1212-0f0f-a0a0-123456789abc
```

1.9.2.2 Example of correct response

Response 200 OK Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 200 OK Body

```
{
  "status" : "OK",
  "description" : "AISP API",
  "version" : "1.1",
  "time": "2020-01-02T05:01:01Z",
  "outage_planned_until": null,
  "poll_interval": "200 ms",
  "details" : {
    "accounts" : {
      "status": "OK",
      "description": "/my/accounts",
      "time": "2020-01-02T05:01:01Z",
    },
    "balances" : {
```



```

        "status": "OK",
        "description": "/my/accounts/{id}/balances",
        "time": "2020-01-02T05:01:01Z",
    },
    "transactions" : {
        "status": "ERROR",
        "description": "/my/accounts/{id}/transactions",
        "time": "2020-01-02T05:01:01Z",
    }
}
}

```

1.9.2.3 Example of correct response with outage status

Response 200 OUTAGE Headers

```

HTTP/1.1 200 OUTAGE
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4

```

Response 200 OUTAGE Body

```

{
  "status" : "OUTAGE",
  "description" : "AISP API",
  "version" : "1.1",
  "time": "2020-01-02T05:01:01Z",
  "outage_planned_until": "2020-02-20T05:01:01Z",
  "poll_interval": "200 ms",
  "details" : {
    "accounts" : {
      "status": "OK",
      "description": "/my/accounts",
      "time": "2020-01-02T05:01:01Z",
    },
  },
}

```

```

    "balances" : {
      "status": "OUTAGE",
      "description": "/my/accounts/{id}/balances",
      "time": "2020-01-02T05:01:01Z",
    },
    "transactions" : {
      "status": "OUTAGE",
      "description": "/my/accounts/{id}/transactions",
      "time": "2020-01-02T05:01:01Z",
    }
  }
}

```

1.9.3 Planned Outages (GET /my/plannedOutages)

It is a universal REST JSON resource that returns information about planned outages or maintenance periods coming up in the future.

Resource provides information for subscribers to know when API services will be out of order.

Planned Outages

A description of the Planned Outages resource with an example.

Resource characteristics

Mandatory:	no
URI:	my/plannedOutages
HTTP Method:	GET
Authorization:	no
Use certificate:	yes
Paging:	no
Sorting:	no
Filtering:	no

Scopes: no

Input character set: no restrictions in relation to chapter 3.1.1

Query parameters of the request: not defined

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

RESPONSE MESSAGE ELEMENTS Planned Outages

LEVEL	MESSAGE ELEMENT	OCCURENCE	FORMAT TYPE	PRESENTATION
+	plannedOutages	[1..1]	±	Planned outages or maintenance periods coming up in the future.
++	createDate	[1..1]	ISODatetime	The date and time when the planned outage was created.
++	startDate	[1..1]	ISODatetime	The beginning of the planned outage.
++	endDate	[1..1]	ISODatetime	End of planned outage.
++	updateAt	[0..1]	ISODatetime	Update in case of planned outage change.
+++	Scope	[1..1]	Array	List of service or services which are affected of planned outages. Possible values: AISP, PISP, CISP, REGISTER
++++	AISP	[0..1]	±	A space-separated list of feature codes affected by the outage for AIS services.
++++	PISP	[0..1]	±	A space-separated list of feature codes affected by the

				outage for PIS services.
++++	CISP	[0..1]	±	A space-separated list of feature codes affected by the outage for CIS services.
++++	REGISTER	[0..1]	±	A space-separated list of feature codes affected by the outage for registration or authentication services.
++	description	[0..1]	Text	Brief description of cause of the resource becoming unavailable.

Service or feature codes that may be affected by the outage

<i>CODE</i>	<i>SCOPE</i>	<i>DESCRIPTION</i>
accountList	AISP	Service to get a list of accounts of a user which is affected of planned outage.
accountBalance	AISP	Service to get account balance which is affected of planned outage.
accountTransactions	AISP	Service to get transaction history which is affected of planned outage.
standingOrders	AISP	Service to get standing orders list, details and awaiting which are affected of planned outage.
paymentOrders	PISP	Service to enter a new payment which is affected of planned outage. Service to get status and detail of a created payment order transaction which is affected of planned outage.
paymentOrders.instantPayments	PISP	Service to enter a new instant payment which is affected of planned outage.
paymentAuthorization	PISP	Service to start the authorization flow of an initiated payment which is affected of planned outage.
standingOrders	PISP	Service to initiate a new standing order payment transaction which is affected of planned outage. Service to get status and detail of a created standing order transaction which is affected of planned outage.

standingOrdersAuthorization	PISP	Service to start the authorization flow of an initiated standing order transaction which is affected of planned outage.
batchPaymentInitiation	PISP	Service to initiate a new batch payment transaction
batchPaymentAuthorization	PISP	Service to start the bulk authorization flow of an initiated batch payment which is affected of planned outage.
registerInitiation	REGISTER	Service to start the application registration flow which is affected of planned outage.
Authorising	REGISTER	Service to start authorisation flow which is affected of planned outage.
issuingToken	REGISTER	Service to get token which is affected of planned outage.
revokeToken	REGISTER	Service to revoke a refresh or access token which is affected of planned outage.
balanceCheck	CISP	Service to get balance check which is affected of planned outage.

1.9.4 Example of request and response for Planned Outages

1.9.4.1 Example of request

Request headers:

```
GET https://api.bank.cz/my/plannedoutages
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 feb 2022 07:21:01 GMT
API-key: 00000000-1212-0f0f-a0a0-123456789abc
```

1.9.4.2 Example of correct response

Response 200 OK Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 200 OK Body

```
{
  "plannedOutages": [
    {
      "createDate": "2022-03-15T08:45:00Z",
      "startDate": "2022-04-01T08:45:00Z",
      "endDate": "2022-04-03T08:45:00Z",
      "updatedAt": "2022-03-26T08:45:00Z",
      "description": "lorem ipsum",
      "scope": {
        "AISP": "accountList accountBalance accountTransactions
StandingOrder",
        "PISP": "paymentOrders paymentAuthorization
batchPaymentsInitiation standingOrders standingOrderAuthorization
batchPaymentAuthorization",
        "CISP": "balanceCheck",
        "REGISTER": "registerInitiation registerData authorising
issuingToken revokeToken"
      }
    },
    {
      "createDate": "2022-03-28T08:45:00Z",
      "startDate": "2022-05-01T08:45:00Z",
      "endDate": "2022-05-03T08:45:00Z",
      "updatedAt": "2022-04-26T08:45:00Z",
      "description": "lorem ipsum",
      "scope": {
        "AISP":
"accountList accountBalance accountTransactions standingOrders",
        "PISP": ""
      }
    }
  ]
}
```

```
]
}
```

1.10 Recommendations for Sandbox PSD2 ASPSP solution features

Sandbox is a non-production part of the solution on which TPP has the opportunity to test the essential integration functions on the PSD2 API issued by ASPSP. Although this is a solution designed primarily for testing, the Sandbox should be operated as a production service mainly in terms of availability. Usually, Sandbox is used as a part of the so-called Developer Portal, where the essential interaction with the developer occurs.

1.10.1 General features of the Sandbox

MANDATORY	OPTIONAL
<ul style="list-style-type: none">the features that can be verified in the Sandbox correspond to the production version of the API or a newer version that will be deployed in production within the next x monthSandbox allows integration (access control) and all documented API services to a minimal extent at the integration level TPP can use Sandbox for testing and integration of PSD2 API. The data in the Sandbox is mostly fictitious (mock data).Sandbox environment is available and accessible (Access should be provided almost immediately through the online form and development portal.)Documentation (Documentation should contain at least specifications, definitions, and also a description of functionalities.)	<ul style="list-style-type: none">Sandbox is freely available for developers without a PSD2 license

1.10.2 Individual functions of PSD2 Sandbox and recommendations for their features

1.10.2.1 Authentication

MANDATORY	OPTIONAL
<ul style="list-style-type: none">• Sandbox supports all supported OAuth2 grant types	<ul style="list-style-type: none">• Sandbox contains a set of pre-generated users. Each user has specified properties (account, balance, number of transactions).• The developer has the opportunity to generate his own user account with specific properties (number of accounts, balance size)• Sandbox allows you to invalidate tokens (as performed by the user in the bank's "Self Service" portal)

1.10.2.2 AIS service

MANDATORY	OPTIONAL
<ul style="list-style-type: none">• Sandbox supports basic PSD2 API<ul style="list-style-type: none">○ for an overview of accounts○ for account information○ about transaction history○ for an overview of standing orders• for all the above endpoints, it is possible to call sample error states (e.g., error code specified in the mock definition, using incorrect values or by a specific user)	<ul style="list-style-type: none">• transaction history is dynamic = values change (e.g., transactions increase)• transaction history contains transactions generated in the PIS service (for combined access)

- transactions in the transaction history are for any period (they are not fixed)

1.10.2.3 PIS service

MANDATORY	OPTIONAL
<ul style="list-style-type: none"> • Sandbox supports basic PSD2 API <ul style="list-style-type: none"> ○ accounts list ○ on payment initiation ○ status of the payment ○ on entering a standing order • for the above endpoints, it is possible to invoke sample error states (e.g., error code specified in the mock definition, using incorrect values or by a specific user) • it is possible to invoke a payment authorization 	<ul style="list-style-type: none"> • initiated payments are entered into the transaction history in the AIS API • the initiated payment will be reflected in the balance change and, for example, in the daily limits • definition of error conditions or specific responses

2 Standard governance

The standard will be changed maximum once a year, suggestions for change may be given by a new mandatory regulation by the bank, a third party through the ČBA or by the working group itself. The suggestion for change must be approved within a proper amendment procedure 6 months before the due date of the changes in force. This implies that the suggestion for a major change must be submitted no later than 1 year before the planned implementation date.

3 Services defined within the standard

DESCRIPTION of three basic API wholes for provision of individual services. The description is broken down by service type, and contains an overview of API resources and an overview of the request elements and response elements of relevant messages, for each service.

Services defined by the standard:

SERVICE	DESCRIPTION
Payment initiation	A service defined as PIS (Payment Initiation Service) by the PSD2 directive
Account information	A service defined as AIS (Account Information Service) by the PSD2 directive
Balance Check	A service defined by the PSD2 directive as information about sufficient funds provided for CISP providers (Card based payment Instrument Issuer Service Provider)

3.1 API Account Information

3.1.1 How to read API account information

The sender of query for API account information must respect permitted characters. Otherwise, the failure to respect them may lead to rejection.

The defined list of enabled characters mainly applies to initiating a payment. For clarification, refer to each resource in the appropriate chapter. In any case, it is necessary to follow the definition in Chapter 1.2.5.

None of these elements may contain separate "/" (slash) at the beginning or end, or two successive slashes in the text. The permitted character set for payment initialisation is based only on the swift character set (that is, exclusively without diacritics - a different character set from the supported characters in CERTIS), i.e. they are the following characters:

```

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

```

Only one query can be sent and processed per call.

All bank account numbers of bank clients in the Czech Republic are defined according to Decree No. 169/2011 Coll. on the rules for the creation of account numbers for making payments. The data element „debtorAccount.identification.iban“ requires an account number in the IBAN format which is defined by the international standard ISO 13616.

Structure of the table of elements:

- **LEVEL** – the character plus „+“ determines the level of element embedding. The basic level is marked „+“ and each another is marked with another character, for instance, the level two is „++“
- **MESSAGE ELEMENT** – element name in the camelCase format
- **OCCURENCE** – describes the occurrence of fields:
 - [1..1] element is mandatory and only occurs once. In the case of a parent element, at least one embedded element must be filled in. If a mandatory element is embedded, the obligation does not pass to the parent element.
 - [0..1] element is optional and occurs only once.
 - [1..n] element is mandatory and occurs n-times. If the value "n" is not numerically defined, the number of repetitions is unlimited.
 - [0..n] element is not mandatory and occurs n-times. If the value "n" is not numerically defined, the number of repetitions is unlimited.
 - [0..0] element is not mandatory and in the specified type (request, attribute or element) does not have to occur, i.e. such element does not have to be implemented by a bank at all.
- **PAYMENT TYPE** – defines for which message types the current element is relevant
- **FORMAT TYPE** – defines the data format. It may be determined by the ISO 20022 standard valid for the item type camt.053 or camt.052 or another standard specifying the data structure (e.g., changes, dates, etc). Some items have a specific required format resulting from the payment system environment in the Czech Republic.
- **PRESENTATION** – contains a general role description

The format type defined as „±“ means the parent element which is then subdivided into other elements.

Conditions for element presence:

- When there is no OR but the parent element has more than 2 levels of elements, elements may be filled in cumulatively
- Parent element after [1..1] – at least one embedded element must be filled in

3.1.2 List of API resources Account Information

Specification of API Information about the account contains a description of resources for the acquisition of information about the payment account/accounts of the bank client.

Overview of resources:

- GET list of client payment accounts

- GET account balance
- GET transaction overview
- GET list of awaiting transactions
- GET list of client's standing orders
- GET standing order detail

3.1.3 List of client payment accounts (GET /my/accounts{?size,page,sort,order})

Paged list of client accounts. Each account contains a unique id usable for URI referencing, e.g. for the account detail.

Resource characteristics

Mandatory: yes

URI: /my/accounts{?size,page,sort,order}

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Account%20information/-getAllAccounts>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: GET

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: yes

Sorting: yes

Filtering: no

Input character set: no restrictions in relation to chapter 3.1.1

Query parameters of the request:

PARAMETER	TYPE	MANDATORY	PURPOSE
size	Number	No	Paging. Number of entries per page
page	Number	No	Paging. Required page. + Default: 0
sort	Text	No	A list of fields separated by comma for sorting, arranged according to the meaning

order	Text	No	A list of arrangement methods (ASC, DESC) separated by comma. The order corresponds to the order of fields in the sort parameter.
--------------	------	----	---

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is always mandatory for this resource

Parameters of the response header:

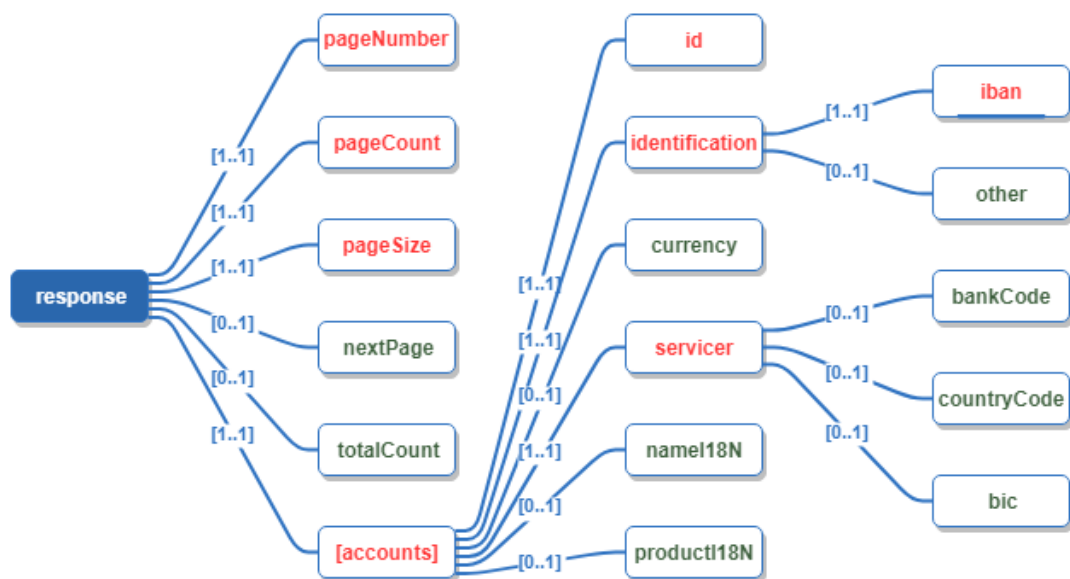
No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of GET response for calling, please see Chapter 3.1.3.1 RESPONSE MESSAGE ELEMENTS List of client payment accounts

Error codes defined for the GET service, List of client payment accounts:

<i>HTTP STATUS CODE</i>	<i>ERROR CODE</i>	<i>PURPOSE</i>
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
401	UNAUTHORISED	Invalid/missing certificate = provider is not authenticated
403	FORBIDDEN	Authentication with an invalid certificate or expired access token, a call that does not match a third-party license.
400	PAGE_NOT_FOUND	Request for a non-existing page
400	PARAMETER_INVALID	The value of the parameter is not valid

3.1.3.1 RESPONSE MESSAGE ELEMENTS List of client payment accounts



LEVEL	MESSAGE ELEMENT	OCCURENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
+	accounts	[1..1]	AISP	±	Set of client accounts
++	id	[1..1]	AISP	Text	API Payment account identifier
++	identification	[1..1]	AISP	±	Payer account identifier
+++	iban	[1..1]	AISP	IBAN2007Identifier	IBAN
+++	other	[0..1]	AISP	Max35Text	Another payer account identifier, e.g. the account number.
++	currency	[0..1]	AISP	CurrencyCode, ISO 4217	Payer account currency
++	servicer	[1..1]	AISP	±	
+++	bankCode	[0..1]	AISP	Text	
+++	countryCode	[0..1]	AISP	CountryCode, ISO 3166 (2 alphanumeric characters code version)	Bank country

+++	bic	[0..1]	AISP	Max35Text	Bank BIC
++	nameI18N	[0..1]	AISP	Text	Account name
++	productI18N	[0..1]	AISP	Text	Product name
++	ownersNames	[0..n]	AISP	Array of Max100Text	Array of name/s of the account holder/s for consumers or trade name/s for commercial clients
++	relationship	[0..1]	AISP	±	
+++	isOwner	[0..1]	AISP	Boolean	True/false if the authenticated person is/is not the owner (payer) of the account
++	suitableScope	[0..1]	AISP	±	
+++	AISP	[0..1]		Text	If nothing is specified, the account supports all AIS services. List of codes that represent services from the table below. Delimiter is space character.
+++	PISP	[0..1]		Text	If nothing is specified, the account supports all PIS services. List of codes that represent services from the table below. Delimiter is space character
+++	CISP	[0..1]		Text	If nothing is specified, the account supports all CIS services. List of codes that represent services from the table below. Delimiter is space character

List of codes for services or functions enabled for a specific payment account

<i>CODE</i>	<i>SCOPE</i>	<i>DESCRIPTION</i>
accountList	AISP	The account appears in the list of accounts.
accountBalance	AISP	It is possible to check the balance of the account using the AIS service.
accountTransactions	AISP	It is possible to get the transaction history of the account using the AIS service.
awaitingTransactions	AISP	
standingOrders	AISP	It is possible to get the list of standing orders of the account using the AIS service.
paymentOrders	PISP	It is possible to initiate a payment using the PIS service on the account.
paymentOrders.instantPayments	PISP	It is possible to initiate a instant payment using the PIS service on the account.
standingOrders	PISP	It is possible to initiate a standing order using the PIS service on the account.
batchPaymentInitiation	PISP	It is possible to initiate a batch payment using the PIS service on the account.
balanceCheck	CISP	For this account, it is possible to request a balance check through the CIS service.

3.1.4 Account balance (GET /my/accounts/{id}/balance{?currency})

The balance of the specific client account according to the reference id account.

Resource characteristics

Mandatory: yes

URI: /my/accounts/{id}/balance{?currency}

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Account%20information/-getAccountsBalances>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP method: GET

Authorization: request **requires** the authorization of user/client as part of the API calling

Use certificate: request **requires** the use of the third-party qualified certificate

Paging: no

Sorting: no

Filtering: no

Input character set: no restrictions in relation to chapter 3.1.1

Path parameters of the request:

PARAMETER	TYPE	MANDATORY	PURPOSE
id	Text	Yes	API payment account identifier from the response to a query on the list of accounts.

Query parameters of the request:

PARAMETER	TYPE	MANDATORY	PURPOSE
currency	Text	No	Required account currency for multi-currency accounts.

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is always mandatory for this resource

Parameters of the response header:

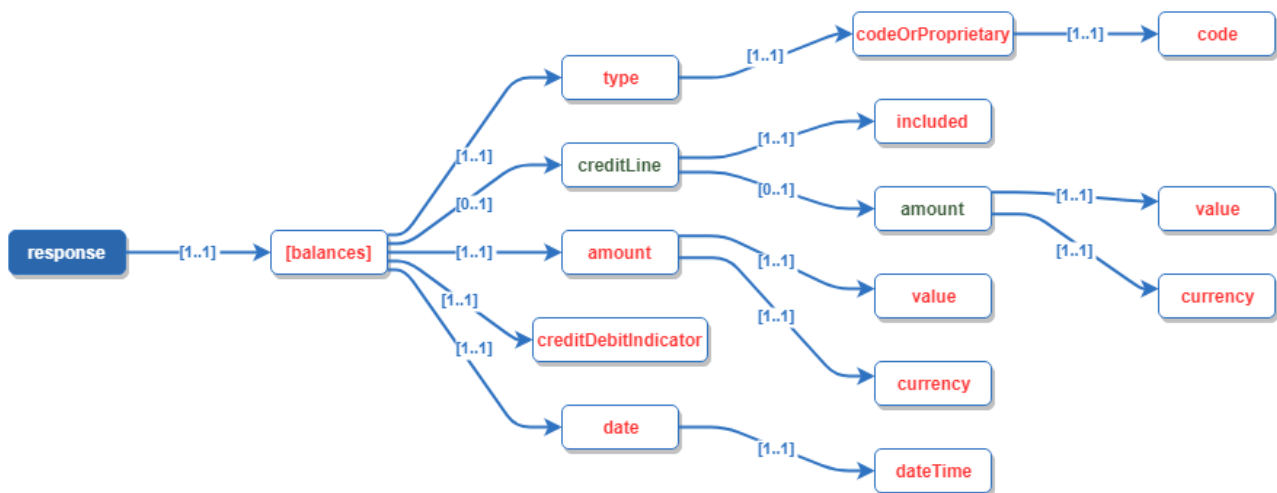
No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of GET response for calling, please see Chapter 3.1.4.1 RESPONSE MESSAGE ELEMENTS Account balance

Error codes defined for the service GET List of client payment accounts:

HTTP STATUS CODE	ERROR CODE	PURPOSE
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
401	UNAUTHORISED	Invalid/missing certificate = provider is not authenticated
404	ID_NOT_FOUND	Invalid or unknown account ID
400	AC09	[InvalidAccountCurrency] – for multicurrency accounts, or currency not supported According to the table of exchanges

3.1.4.1 RESPONSE MESSAGE ELEMENTS Account balance



LEVEL	MESSAGE ELEMENT	OCCURENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
+	balances	[1..1]	AISP	±	Set of balances of client payment account
++	type	[1..1]	AISP	±	Determines the balance type to which

					Information on account balance applies
+++	codeOrProprietary	[1..1]	AISP	±	
++++	code	[1..1]	AISP	Type of balance	Codes of balance types see chapter code (the balance type in the account balance response)
++	creditLine	[0..1]	AISP	±	Amount of agreed permitted debit/overdraft
+++	included	[1..1]	AISP	Boolean	
+++	amount	[0..1]	AISP	±	Amount of agreed permitted debit
++++	value	[1..1]	AISP	Number	Amount of agreed permitted debit
++++	currency	[1..1]	AISP	Text	The currency corresponds to the currency of the account to which the balance information is generated
++	amount	[1..1]	AISP	±	Value/amount of account balance according to balance type. The currency corresponds to the currency of the account to which the balance information is generated
+++	value	[1..1]	AISP	Number	Amount of account balance
+++	currency	[1..1]	AISP	Text	The currency corresponds to the currency of the account to which the balance information is generated
++	creditDebitIndicator	[1..1]	AISP	Text	Indication of whether the balance in the account for which the

					balance information is generated
++	date	[1..1]	AISP	±	Date (and time) of balance of the account for which the balance information is generated
+++	dateTime	[1..1]	AISP	Text	Date or date and time of the balance according to ISO 8601

3.1.5 Overview of transactions (GET

/my/accounts/{id}/transactions{?fromDate,toDate,currency,size,page,sort,order})

Paged list of transactions of a selected client account.

Resource characteristics

Mandatory: yes

URI: /my/accounts/{id}/transactions{?fromDate,toDate,currency,size,page,sort,order}

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Account%20information/-getAccountsTransactions>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: GET

Authorization: request **requires** the authorization of user/client as part of the API calling

Use certificate: request **requires** the use of the third-party qualified certificate

Paging: yes

Sorting: yes

Filtering: no

Input character set: no restrictions in relation to chapter 3.1.1

Path parameters of the request:

PARAMETER	TYPE	MANDATORY	PURPOSE
-----------	------	-----------	---------

id	Text	Yes	API payment account identifier from the response to a query on the list of accounts.
-----------	------	-----	--

Query parameters of the request:

<i>PARAMETER</i>	<i>TYPE</i>	<i>MANDATORY</i>	<i>PURPOSE</i>
fromDate	Text	No	Date and time of the start of required transaction history
toDate	Text	No	Date and time of the end of required transaction history [inclusive]
currency	Text	No	Required account currency for multi-currency accounts
size	Number	No	Paging. Number of entries per page
page	Number	No	Paging. Required page. + Default: 0
sort	Text	No	A list of fields separated by comma for sorting, arranged according to the meaning
order	Text	No	A list of arrangement methods (ASC, DESC) separated by comma. The order corresponds to the order of fields in the sort parameter.

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is always mandatory for this resource

Parameters of the response header:

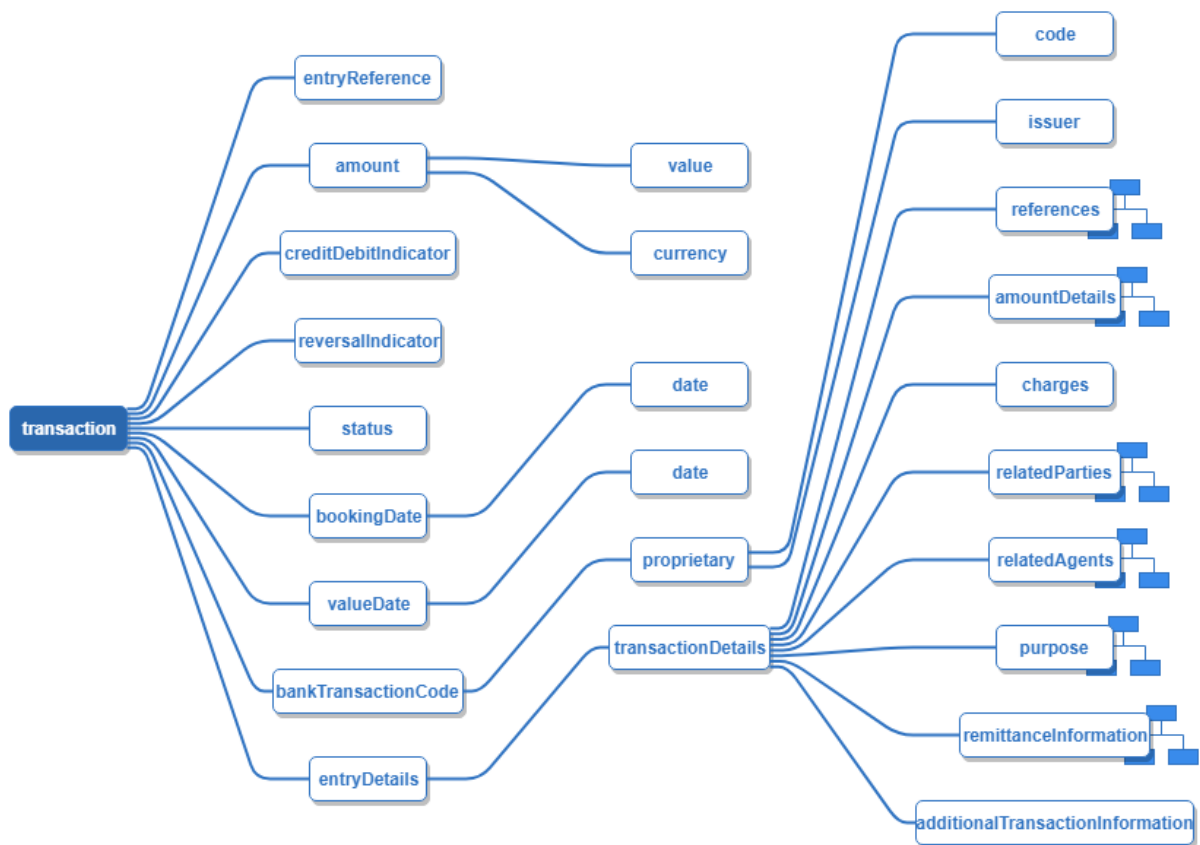
No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of GET response for calling, please see Chapter 3.1.5.1 RESPONSE MESSAGE ELEMENTS Overview of transactions

Error codes defined for the GET service List of client payment accounts:

HTTP STATUS CODE	ERROR CODE	PURPOSE
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
401	UNAUTHORISED	Invalid/missing certificate = provider is not authenticated
404	ID_NOT_FOUND	Invalid or unknow account ID
404	PAGE_NOT_FOUND	Query for not existing page
400	PARAMETER_INVALID	Parameter value is not valid
400	AC09	[InvalidAccountCurrency] – for multicurrency accounts, or currency not supported according to the table of exchanges
400	DT01	[InvalidDate] Invalid date

3.1.5.1 RESPONSE MESSAGE ELEMETS Overview of transactions



LEVEL	MESSAGE ELEMENT	FORMAT TYPE	PRESENTATION
+	entryReference	Max35Text	Identification no. of the payment assigned by the bank.
+	amount	±	Payment amount in the account currency for which payment history is generated.
++	value	Amount	
++	currency	CurrencyCode	
+	creditDebitIndicator	CreditDebitCode	Indication of whether it is a debit payment or credit payment of the account. An expression of whether it is a debit or credit payment is marked with one of the codes below: DBIT: to the debit of CRDT: In other cases
+	reversalIndicator	TrueFalseIndicator	Indication of whether it is a cancellation. An expression of whether it is a cancellation or not is marked with one of the codes below: true: It is cancellation false: It is not cancellation
+	status	Code	Item status (debited or credited payments) of the account from the point of view of the bank. The statement will present only posted items, with the constant BOOK, or blocked items, with the constant PDNG.

+	bookingDate	±	Date of processing/posting of payment by bank in the format ISODate, or ISODateTime, i.e. YYYY-MM-DD, or YYYY-MM-DDThh:mm:ss.STZD.
++	date	ISODate ISODateTime	<p>Date of processing/posting of payment by bank in the format ISODate, or ISODateTime, i.e. YYYY-MM-DD, or YYYY-MM-DDThh:mm:ss.STZD, depending on the transaction type and method how the bank presents data (and time) of payment processing/posting. Mainly for card or cash transactions it is posted as ISODateTime.</p> <p>where:</p> <ul style="list-style-type: none"> YYYY = four-digit year MM = two-digit month (01=January, etc.) DD = two-digit day of month (01 through 31) hh = two digits of hour (00 through 23) (am/pm NOT allowed) mm = two digits of minute (00 through 59) ss = two digits of second (00 through 59) TZD = time zone designator (Z or +hh:mm or -hh:mm)

+	valueDate	±	Due date/payment foreign currency in the format ISODate, or ISODateTime, i.e. YYYY-MM-DD, or YYYY-MM-DDThh:mm:ss.sTZD.
++	date	ISODate/ISODateTime	<p>Due date/payment foreign currency in the format ISODate, , or ISODateTime, i.e. YYYY-MM-DD, or YYYY-MM-DDThh:mm:ss.sTZD, depending on the transaction type and method how the bank presents data (and time) of due date/ payment foreign currency. Mainly for card or cash transactions it is posted as ISODateTime.</p> <p>where:</p> <ul style="list-style-type: none"> YYYY = four-digit year MM = two-digit month (01=January, etc.) DD = two-digit day of month (01 through 31) hh = two digits of hour (00 through 23) (am/pm NOT allowed) mm = two digits of minute (00 through 59) ss = two digits of second (00 through 59) TZD = time zone designator (Z or +hh:mm or -hh:mm) <p>where:</p> <ul style="list-style-type: none"> YYYY = four-digit year MM = two-digit month (01=January, etc.) DD = two-digit day of month (01 through 31).

+	holdExpiration	±	Date when card transaction hold expire in the format ISODate. The element will present only in status code with the constant PDNG.
++	date	ISODate	The date when a card transaction hold will be expire. Format YYYY-MM-DD.
+	bankTransactionCode	±	The code of banking transaction according to the code list of the Czech Banking Association assigned to a specific payment. Each bank uses an own code list to identify payments, which is, however, based on the 1st to 3rd level of transaction code list according to the CBA Standard for camt.053.
++	proprietary	±	The code of banking transaction according to the code list of the Czech Banking Association assigned to a specific payment. Each bank uses an own code list to identify payments, which is, however, based on the 1st to 3rd level of transaction code list according to the CBA Standard for camt.053.

+++	code	Max35Text	The code of banking transaction according to the code list of the Czech Banking Association assigned to a specific payment. Each bank uses an own code list to identify payments, which is, however, based on the 1st to 3rd level of transaction code list according to the CBA Standard for camt.053.
+++	issuer	Max35Text	Identification of the code list issuer for banking transactions which acquires the value of the CBA.
+	entryDetails	±	Turnover details. The level repeats only for the given item
++	transactionDetails	±	Payment details. The level repeats only for the given item.
+++	references	±	A set of references unambiguously identifying the payment.
++++	messageIdentification	Max35Text	Assumed payment identification entered by the client on its initiation or order of the payment in the payment history.
++++	paymentId	Max255Text	Id of initiated payment. The identifier of the payment from the PISP service

++++	accountServicerReference	Max35Text	A banking reference assigned to the payment, e.g., on initiation through direct banking services.
++++	paymentInformationIdentification	Max35Text	Another/next banking reference assigned to the payment assigned by the bank, for card payments, the card sequence number may be entered, or the specific symbol may be entered here.
++++	instructionIdentification	Max35Text	Payment identification entered by a third party, or a constant symbol may be entered.
++++	endToEndIdentification	Max35Text	A unique identifier specified by the client initiating the payment that serves for unambiguous payment identification and is passed in the unchanged state throughout the payment chain, or, there may be a variable symbol filled in here.
++++	mandateIdentification	Max35Text	For SEPA collections, Unique Mandate Reference stated for the given SEPA collection, as a mandatory field [1..1].

++++	chequeNumber	Max35Text	For cheque transactions there may be a cheque number; for card transactions there may be a card number with asterisks. The card no. is each time in the format xxxxxxxxxxx1234 where according to the standard, only 0-9 may be used (8-28 char.) i.e. without illustrative xxxx.
++++	clearingSystemReference	Max35Text	A bank-defined code list value identifying the type of payment or payment type used. For card transactions, card association identification may be indicated.
+++	amountDetails	±	Details of the payment amount, especially if it is a conversion payment or cashback.
++++	instructedAmount	±	Amount and currency of payment in the currency that was required by the client for transfer. E.g., for intra-bank payments, the payer account currency and the payment currency if the client required the payment in the payer account currency.
+++++	amount	±	The original amount and currency of payment that was required by the client for transfer.
+++++	value	Amount	
+++++	currency	CurrencyCode	

++++	transactionAmount	±	The amount and currency of payment for cumulated payments and Cashback.
+++++	amount	±	The amount and currency of payment for cumulated payments and Cashback, where the total amount of payment including the part for Cashback is entered here.
+++++	value	Amount	
+++++	currency	CurrencyCode	
++++	counterValueAmount	±	The amount and currency of payment in the client account currency after conversion of amount that was required by the client for transfer.
+++++	amount	±	The final amount and payment currency that was required by the client for transfer.
+++++	value	Amount	
+++++	currency	CurrencyCode	
+++++	currencyExchange	±	Information on currencies and exchange rates used.
+++++	sourceCurrency	CurrencyCode	Client account currency (source/original currency/payer account currency for intrabank conversion payments).
+++++	targetCurrency	CurrencyCode	Payment currency (End/target currency/payee account currency for intrabank conversion payments).

+++++	exchangeRate	BaseOneRate	The exchange rate used to charge a payment. Only one rate is filled in, even in the case of cross conversion.
++++	proprietaryAmount	±	Amount of cash withdrawal through the Cashback service.
+++++	type	Max35Text	The constant "CASHBACK" is to be filled in.
+++++	amount	±	Cashback amount and currency – only the part of total payment, to which the cash amount withdrawn falls within the Cashback service.
+++++	value	Amount	
+++++	currency	CurrencyCode	
+++	charges	±	Information on fees.
++++	bearer	Code	Indication of that payments will be charged to his debit (OUR, SHA, BEN).
+++	relatedParties	±	Information on the payer, the payer's account and the original payer and payee, the payee's account and the final payee in the payment.
++++	debtor	±	Information on the payer. It is according to the payment direction, and filled in in the case of counterparty.
+++++	name	Max70Text	Payer name.
+++++	postalAddress	±	Payer postal address.
+++++	streetName	Max70Text	Street name used for the payer postal address.

++++++	buildingNumber	Max16Text	Land registry no. used for the payer postal address.
++++++	postCode	Max16Text	Postcode used for the payer postal address.
++++++	townName	Max35Text	Town name used for the payer postal address.
++++++	country	CountryCode, ISO 3166 (2 alphanumeric characters code version)	Country name used for the payer postal address.
++++++	addressLine	Max70Text	Unstructured record of payer postal address.
+++++	identification	±	Payer identification.
++++++	organisationIdentification	±	Clear payer identification as the organization/legal entity.
+++++++	bicOrBei	BICIdentifier	Payer identification as the organization/legal entity in the form of BIC or BEI code.
+++++++	other	±	Other payer identification as the organization/legal entity.
+++++++	identification	Max35Text	Other payer identification as the organization/legal entity in unstructured form.
+++++++	schemeName	±	Code type for payer identification as the organization/legal entity.

+++++++	code	Code	Code type for payer identification as the organization/legal entity in the form of code according to the ISO code list.
+++++++	proprietary	Max35Text	Code type for payer identification as the organization/legal entity in the free text format.
+++++++	issuer	Max35Text	Code issuer for the payer identification as the organization/legal entity.
+++++	privateIdentification	±	Clear payer identification as the natural person.
+++++	other	±	Another payer identification as the natural person.
+++++++	identification	Max35Text	Other payer identification as the natural person in unstructured form.
+++++++	schemeName	±	Code type for payer identification as the natural person.
+++++++	code	Code	Code type for payer identification as the natural person in the form of code according to the ISO code list.
+++++++	proprietary	Max35Text	Code type for payer identification as the natural person in free text format.
+++++++	issuer	Max35Text	Code issuer for payer identification as the natural person.

++++	debtorAccount	±	Information on the account payer. It is according to the payment direction, and filled in in the case of counterparty.
+++++	identification	±	Identification of account payer type.
+++++	iban	IBAN2007Identifier	Payer account no. in the international format of IBAN account no.
+++++	other	±	Payer account no. in other/local format of account no.
+++++	identification	Max34Text	The value of payer account no. in other/local format of account no.
+++++	currency	CurrencyCode	Payer account currency.
+++++	name	Max70Text	Payer account name.
++++	ultimateDebtor	±	Information on the original/actual payer. It is according to the payment direction, and filled in in the case of counterparty.
+++++	name	Max70Text	Name of the original/actual payer.
+++++	postalAddress	±	Postal address of the original payer.
+++++	streetName	Max70Text	Street name used for the payer postal address of the original payer.
+++++	buildingNumber	Max16Text	Land registry no. used for the original payer postal address.
+++++	postCode	Max16Text	Postcode used for the original payer postal address.

++++++	townName	Max35Text	Town name used for the original payer postal address.
++++++	country	CountryCode, ISO 3166 (2 alphanumeric characters code version)	Country name used for the original payer postal address.
++++++	addressLine	Max70Text	Unstructured record of original payer postal address.
+++++	identification	±	Original payer identification.
++++++	organisationIdentification	±	Clear identification of the original payer as the organization/legal entity.
+++++++	bicOrBei	BICIdentifier	Identification of the original payer as the organization/legal entity in the form of BIC or BEI code.
+++++++	other	±	Other identification of the original payer as the organization/legal entity.
+++++++	identification	Max35Text	Other identification of the original payer as the organization/legal entity in unstructured form.
+++++++	schemeName	±	Code type for the identification of the original payer as the organization/legal entity.

+++++++	code	Code	Code type for the identification of the original payer as the organization/legal entity in the form of code according to the ISO code list.
+++++++	proprietary	Max35Text	Code type for the identification of the original payer as the organization/legal entity in free text format.
+++++++	issuer	Max35Text	Code issuer for the original payer identification as the organization/legal entity.
+++++	privateIdentification	±	Clear identification of the original payer as the natural person.
+++++	other	±	Other identification of the original payer as the natural person.
+++++++	identification	Max35Text	Other identification of the original payer as the natural person in unstructured form.
+++++++	schemeName	±	Code type for the identification of the original payer as the natural person.
+++++++	code	Code	Code type for the identification of the original payer as the natural person in the form of code according to the ISO code list.
+++++++	proprietary	Max35Text	Code type for the identification of the original payer as the natural person in the free

			text format.
+++++++	issuer	Max35Text	Code issuer for the identification of the original payer as the natural person.
++++	creditor	±	Information on the payee. It is according to the payment direction, and filled in in the case of counterparty.
+++++	name	Max70Text	Payee name.
+++++	postalAddress	±	Payee postal address.
+++++	streetName	Max70Text	Street name used for the payee postal address.
+++++	buildingNumber	Max16Text	Land registry no. used for the payee postal address.
+++++	postCode	Max16Text	Postcode used for the payee postal address.
+++++	townName	Max35Text	Town name used for the payee postal address.
+++++	country	CountryCode, ISO 3166 (2 alphanumeric characters code version)	Country name used for the payee postal address.
+++++	addressLine	Max70Text	Unstructured record of the payee postal address.
+++++	identification	±	Payee identification.
+++++	organisationIdentification	±	Clear identification of the payee as the organization/legal entity.

+++++++	bicOrBei	BICIdentifier	Payee identification as the organization/legal entity in the form of BIC or BEI code.
+++++++	other	±	Other identification of the payee as the organization/legal entity.
+++++++	identification	Max35Text	Other identification of the payee as the organization/legal entity in unstructured form.
+++++++	schemeName	±	Code type for the payee identification as the organization/legal entity.
+++++++	code	Code	Code type for the payee identification as the organization/legal entity in the form of code according to the ISO code list.
+++++++	proprietary	Max35Text	Code type for the payee identification as the organization/legal entity in the free text format
+++++++	issuer	Max35Text	Code issuer for the payee identification as the organization/legal entity.
+++++	privateIdentification	±	Clear identification of the payee as the natural person.
+++++	other	±	Other identification of the payee as the natural person.

+++++++	identification	Max35Text	Other identification of the payee as the natural person in unstructured form.
+++++++	schemeName	±	Code type for the payee identification as the natural person.
+++++++	code	Code	Code type for the payee identification as the natural person in the form of code according to the ISO code list.
+++++++	proprietary	Max35Text	Code type for the payee identification as the natural person in the free text format.
+++++++	issuer	Max35Text	Code issuer for the payee identification as the natural person.
++++	creditorAccount	±	Information on the payee account. It is according to the payment direction, and filled in in the case of counterparty.
+++++	identification	±	Identification of payee account type.
+++++	iban	IBAN2007Identifier	Payee account no. in the international format of IBAN account no.
+++++	other	±	Payee account no. in other/local format of account no.
+++++	identification	Max34Text	The value of payee account no. in other/local format of account no.
+++++	currency	CurrencyCode	Payee account currency.
+++++	name	Max70Text	Payee account name.

++++	ultimateCreditor	±	Information on the final payee. It is according to the payment direction, and filled in in the case of counterparty.
+++++	name	Max70Text	Final payee name.
+++++	postalAddress	±	Final payee postal address.
+++++	streetName	Max70Text	Street name used for the final payee postal address.
+++++	buildingNumber	Max16Text	Land registry no. used for the final payee postal address.
+++++	postCode	Max16Text	Postcode used for the final payee postal address.
+++++	townName	Max35Text	Town name used for the final payer postal address.
+++++	country	CountryCode, ISO 3166 (2 alphanumeric characters code version)	Country name used for the final payer postal address.
+++++	addressLine	Max70Text	Unstructured record of final payee postal address.
+++++	identification	±	Final payee identification.
+++++	organisationIdentification	±	Clear identification of the final payee as the organization/legal entity.
+++++	bicOrBei	BICIdentifier	Identification of the final payee as the organization/legal entity in the form of the BIC or BEI code.

+++++++	other	±	Other identification of the final payee as the organization/legal entity.
+++++++	identification	Max35Text	Other identification of the final payee as the organization/legal entity in unstructured form.
+++++++	schemeName	±	Code type for the identification of final payee as the organization/legal entity.
+++++++	code	Code	Code type for the identification of final payee as the organization/legal entity in the form of code according to the ISO code list.
+++++++	proprietary	Max35Text	Code type for the identification of final payee as the organization/legal entity in free text format.
+++++++	issuer	Max35Text	Code issuer for the identification of the final payee as the organization/legal entity.
+++++	privateIdentification	±	Clear identification of the final payee as the natural person.
+++++	other	±	Other identification of the final payee as the natural person.

+++++++	identification	Max35Text	Other identification of the final payee as the natural person in unstructured form.
+++++++	schemeName	±	Code type for the identification of the final payee as the natural person.
+++++++	code	Code	Code type for the identification of the final payee as the natural person in the form of code according to the ISO code list.
+++++++	proprietary	Max35Text	Code type for the identification of the final payee as the natural person in the free text format.
+++++++	issuer	Max35Text	Code issuer for the identification of the final payee as the natural person.
++++	proprietary	±	For more detailed identification of on which ATM the card transaction occurred.
++++	type	Max35Text	For identification of whether for the card transaction it is own / external ATM.
++++	party	±	For identification of the ATM name/owner.
++++	name	Max140Text	For identification of ATM name/location.
+++	relatedAgents	±	Information on the payer bank and the payee bank in the payment.

++++	debtorAgent	±	Information on the payer bank. It is according to the payment direction, and filled in in the case of counterparty.
+++++	financialInstitutionIdentification	±	Payer bank code in the international format BIC / SWIFT code.
+++++	bic	BICIdentifier	BIC / SWIFT code of the payer bank
+++++	clearingSystemMemberIdentification	±	Local format of the payer bank code – either in the code form or in the form of text description.
+++++	clearingSystemIdentification	±	Payer bank identification in the local payment system in which the payer bank works.
+++++	code	Code	Payer bank identification in the local payment system in which the payer bank works in the form of payment system code.
+++++	proprietary	Max35Text	Payer bank identification in the local payment system in which the payer bank works in unstructured form of descriptive text.
+++++	memberIdentification	Max35Text	Local code format of the payer bank code.
+++++	name	Max70Text	Payer bank code as stated in the code list of world banks (SWIFT Directory).
+++++	postalAddress	±	Postal address of the payer bank.

+++++++	streetName	Max70Text	Street name used for the payer postal address.
+++++++	buildingNumber	Max16Text	Land registry no. used for the postal address of the payer's bank.
+++++++	postCode	Max16Text	Postcode used for the payer's bank postal address.
+++++++	townName	Max35Text	Town name used for the payer's bank postal address.
+++++++	country	CountryCode, ISO 3166 (2 alphanumeric characters code version)	Country name used for the payer's bank postal address.
+++++++	addressLine	Max70Text	Unstructured record of payer's bank postal address.
+++++	other	±	Other payee bank identification.
+++++++	identification	Max35Text	Other payer bank identification, mainly in the form of the so-called local bank code.
++++	creditorAgent	±	Information on the payee bank. It is according to the payment direction, and filled in in the case of counterparty.
+++++	financialInstitutionIdentification	±	Code of the payee bank in the international format BIC / SWIFT code.
+++++	bic	BICIdentifier	BIC / SWIFT code of the payee bank
+++++	clearingSystemMemberIdentification	±	Local format of the payee bank code - either in the code form or in the form of text description.

+++++++	clearingSystemIdentification	±	Payee bank identification in the local payment system in which the payee bank works.
+++++++	code	Code	Payee bank identification in the local payment system in which the payee bank works in the form of payment system code.
+++++++	proprietary	Max35Text	Payee bank identification in the local payment system in which the payee bank works in unstructured form of descriptive text.
+++++++	memberIdentification	Max35Text	Local code format of the payee bank code.
+++++++	name	Max70Text	Payee bank name.
+++++++	postalAddress	±	Payee's bank postal address.
+++++++	streetName	Max70Text	Street name used for payee's bank postal address.
+++++++	buildingNumber	Max16Text	Land registry no. used for the payee's bank postal address.
+++++++	postCode	Max16Text	Postcode used for the payee's bank postal address.
+++++++	townName	Max35Text	Town name used for the payee's bank postal address.
+++++++	country	CountryCode, ISO (2 alphanumeric characters code version)	Country name used for the payee's bank postal address.

+++++++	addressLine	Max70Text	Unstructured record of the payee's bank postal address.
+++++	other	±	Other payee bank identification, mainly in the form of the so-called local bank code.
+++++++	identification	Max35Text	Information on the payee bank. It is according to the payment direction, and filled in in the case of counterparty.
+++	purpose	±	Payment purpose.
++++	code	Code	Payment purpose expressed as a code that was entered in the payment.
++++	proprietary	Max35Text	Payment purpose expressed as unstructured information that was entered in the payment.
+++	remittanceInformation	±	Additional payment information.
++++	unstructured	Max140Text	Additional payment information filled in in the payment as unstructured information. If the payment states several repetitions of unstructured record supplementing payment information, then only the first of them will be considered.

++++	structured	±	<p>The completion of payment symbols (variable, specific and constant) is expected.</p> <p>If the payment does not have a variable, specific or constant symbol, then the entire Structured Remittance Information will be left empty.</p>
+++++	creditorReferenceInformation	±	To display information on variable, specific and constant symbol.
+++++	reference	Max35Text	<p>Start the field with the VS code: (for variable symbol), or SS: (for specific symbol) or KS: (for constant symbol). In one repetition of the reference field, all three symbols may be stated, each time maximum one repetition of each individual symbol. The method of recording symbols is VS: KS: SS:)[0-9]{1,10}.</p>
+++	additionalTransactionInformation	Max500Text	<p>Additional information provided by the bank. Only stated if it is included in the record. Additional info for SEPA DD is entered here (e.g., Creditor Identifier, Payment scheme, SEPA collection order, etc.)</p>

+++	additionalRemittanceInformation	Max140Text	Additional free text information for payer of up to 140 characters.
-----	---------------------------------	------------	---

3.1.5.1 MESSAGE ELEMENTS Overview of transactions - occurrence in individual payment types

3.1.5.1.1 Interest – occurrence of elements

LEVEL	MESSAGE ELEMENT	OCCURRENCE
+	entryReference	[0..1]
+	amount	[1..1]
++	value	[1..1]
++	currency	[1..1]
+	creditDebitIndicator	[1..1]
+	reversalIndicator	[0..1]
+	status	[1..1]
+	bookingDate	[1..1]
++	date	[1..1]
+	valueDate	[1..1]
++	date	[1..1]
+	bankTransactionCode	[1..1]
++	proprietary	[1..1]
+++	code	[1..1]
+++	issuer	[1..1]

+	entryDetails	[0..1]
++	transactionDetails	[0..1]
+++	references	[0..1]
++++	paymentId	[0..1]
++++	messageIdentification	[0..1]
++++	accountServicerReference	[0..1]
++++	paymentInformationIdentification	[0..1]
++++	instructionIdentification	[0..1]
++++	endToEndIdentification	[0..1]
++++	chequeNumber	[0..1]
++++	clearingSystemReference	[0..1]
+++	amountDetails	[0..1]
++++	instructedAmount	[0..1]
+++++	amount	[1..1]
+++++	value	[1..1]
+++++	currency	[1..1]
++++	transactionAmount	[0..0]
+++++	amount	[1..1]
+++++	value	[1..1]
+++++	currency	[1..1]
++++	counterValueAmount	[0..1]
+++++	amount	[1..1]
+++++	value	[1..1]
+++++	currency	[1..1]
++++	currencyExchange	[0..1]
+++++	sourceCurrency	[1..1]
+++++	targetCurrency	[0..1]
+++++	exchangeRate	[1..1]
++++	proprietaryAmount	[0..0]
+++++	type	[0..0]

+++++	amount	[0..0]
+++++	value	[1..1]
+++++	currency	[1..1]
+++	relatedParties	[0..1]
++++	debtor	[0..1]
+++++	name	[0..1]
++++	debtorAccount	[0..1]
+++++	identification	[1..1]
+++++	iban	[1..1]..either or other identification
+++++	other	[1..1].. either or IBAN
+++++	identification	[1..1]
++++	creditor	[0..1]
+++++	name	[0..1]
++++	creditorAccount	[0..1]
+++++	identification	[1..1]
+++++	iban	[1..1].. either or other identification
+++++	other	[1..1].. either or IBAN
+++++	identification	[1..1]
+++	relatedAgents	[0..1]
++++	debtorAgent	[0..1]
+++++	financialInstitutionIdentification	[1..1]
+++++	bic	[1..1]...either or other identification
+++++	name	[0..1]
+++++	other	[1..1]... either or BIC
+++++	identification	[1..1]
++++	creditorAgent	[0..1]
+++++	financialInstitutionIdentification	[1..1]
+++++	bic	[1..1]... either or other

		identification
+++++	name	[0..1]
+++++	other	[1..1]... either or BIC
+++++	identification	[1..1]
+++	remittanceInformation	[0..1]
++++	unstructured	[0..1]
++++	structured	[0..1]
+++++	creditorReferenceInformation	[0..1]
+++++	reference	[0..1]
+++	additionalTransactionInformation	[0..1]

3.1.5.1.2 Fee – occurrence of elements

LEVEL	MESSAGE ELEMENT	OCCURRENCE
+	entryReference	[0..1]
+	amount	[1..1]
++	value	[1..1]
++	currency	[1..1]
+	creditDebitIndicator	[1..1]
+	reversalIndicator	[0..1]
+	status	[1..1]
+	bookingDate	[1..1]
++	date	[1..1]
+	valueDate	[1..1]
++	date	[1..1]
+	bankTransactionCode	[1..1]
++	proprietary	[1..1]
+++	code	[1..1]

+++	issuer	[1..1]
+	entryDetails	[0..1]
++	transactionDetails	[0..1]
+++	references	[0..1]
++++	paymentId	[0..1]
++++	messageIdentification	[0..1]
++++	accountServicerReference	[0..1]
++++	paymentInformationIdentification	[0..1]
++++	instructionIdentification	[0..1]
++++	endToEndIdentification	[0..1]
++++	chequeNumber	[0..1]
++++	clearingSystemReference	[0..1]
+++	amountDetails	[0..1]
++++	instructedAmount	[0..1]
+++++	amount	[1..1]
+++++	value	[1..1]
+++++	currency	[1..1]
++++	transactionAmount	[0..0]
+++++	amount	[1..1]
+++++	value	[1..1]
+++++	currency	[1..1]
++++	counterValueAmount	[0..1]
+++++	amount	[1..1]
+++++	value	[1..1]
+++++	currency	[1..1]
++++	currencyExchange	[0..1]
+++++	sourceCurrency	[1..1]
+++++	targetCurrency	[0..1]
+++++	exchangeRate	[1..1]
++++	proprietaryAmount	[0..0]

+++++	type	[0..0]
+++++	amount	[0..0]
+++++	value	[1..1]
+++++	currency	[1..1]
+++	relatedParties	[0..1]
++++	debtor	[0..1]
+++++	name	[0..1]
++++	debtorAccount	[0..1]
+++++	identification	[1..1]
+++++	iban	[1..1]..either or other identification
+++++	other	[1..1]..either or IBAN
+++++	identification	[1..1]
++++	creditor	[0..1]
+++++	name	[0..1]
++++	creditorAccount	[0..1]
+++++	identification	[1..1]
+++++	iban	[1..1]..either or other identification
+++++	other	[1..1]..either or IBAN
+++++	identification	[1..1]
+++	relatedAgents	[0..1]
++++	debtorAgent	[0..1]
+++++	financialInstitutionIdentification	[1..1]
+++++	bic	[0..1]...either or other identification
+++++	name	[1..1]
+++++	other	[1..1]...either or BIC
+++++	identification	[1..1]
++++	creditorAgent	[0..1]
+++++	financialInstitutionIdentification	[1..1]

++++++	bic	[1..1]...either or other identification
++++++	name	[0..1]
++++++	other	[1..1]...either or BIC
++++++	identification	[1..1]
+++	remittanceInformation	[0..1]
++++	unstructured	[0..1]
++++	structured	[0..1]
+++++	creditorReferenceInformation	[0..1]
+++++	reference	[0..1]
+++	additionalTransactionInformation	[0..1]

3.1.5.1.3 Domestic payment – occurrence of elements

LEVEL	MESSAGE ELEMENT	OCCURRENCE
+	entryReference	[0..1]
+	amount	[1..1]
++	value	[1..1]
++	currency	[1..1]
+	creditDebitIndicator	[1..1]
+	reversalIndicator	[0..1]
+	status	[1..1]
+	bookingDate	[1..1]
++	date	[1..1]
+	valueDate	[1..1]
++	date	[1..1]
+	bankTransactionCode	[1..1]
++	proprietary	[1..1]
+++	code	[1..1]

+++	issuer	[1..1]
+	entryDetails	[1..1]
++	transactionDetails	[1..1]
+++	references	[0..1]
++++	paymentId	[0..1]
++++	messageIdentification	[0..1]
++++	accountServicerReference	[0..1]
++++	paymentInformationIdentification	[0..1]
++++	instructionIdentification	[0..1]
++++	endToEndIdentification	[0..1]
++++	chequeNumber	[0..1]
++++	clearingSystemReference	[0..1]
+++	amountDetails	[0..1]
++++	instructedAmount	[0..1]
+++++	amount	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
++++	transactionAmount	[0..0]
+++++	amount	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
++++	counterValueAmount	[0..1]
+++++	amount	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
+++++	currencyExchange	[0..1]
++++++	sourceCurrency	[1..1]
++++++	targetCurrency	[0..1]
++++++	exchangeRate	[1..1]
++++	proprietaryAmount	[0..0]

+++++	type	[0..0]
+++++	amount	[0..0]
+++++	value	[1..1]
+++++	currency	[1..1]
+++	relatedParties	[1..1]
++++	debtor	[0..1]
+++++	name	[0..1]
++++	debtorAccount	[1..1]
+++++	identification	[1..1]
+++++	iban	[1..1]..either or other identification
+++++	other	[1..1].. either or IBAN
+++++	identification	[1..1]
++++	creditor	[0..1]
+++++	name	[0..1]
++++	creditorAccount	[1..1]
+++++	identification	[1..1]
+++++	iban	[1..1].. either or other identification
+++++	other	[1..1].. either or IBAN
+++++	identification	[1..1]
+++	relatedAgents	[1..1]
++++	debtorAgent	[1..1]
+++++	financialInstitutionIdentification	[1..1]
+++++	bic	[1..1]...either or other identification
+++++	name	[0..1]
+++++	other	[1..1]...either or BIC
+++++	identification	[1..1]
++++	creditorAgent	[1..1]
+++++	financialInstitutionIdentification	[1..1]

++++++	bic	[1..1]...either or other identification
++++++	name	[0..1]
++++++	other	[1..1]...either or BIC
++++++	identification	[1..1]
+++	remittanceInformation	[0..1]
++++	unstructured	[0..1]
++++	structured	[0..1]
+++++	creditorReferenceInformation	[0..1]
+++++	reference	[0..1]
+++	additionalTransactionInformation	[0..1]

3.1.5.1.4 SEPA payment – occurrence of elements

LEVEL	MESSAGE ELEMENT	OCCURRENCE
+	entryReference	[0..1]
+	amount	[1..1]
++	value	[1..1]
++	currency	[1..1]
+	creditDebitIndicator	[1..1]
+	reversalIndicator	[0..1]
+	status	[1..1]
+	bookingDate	[1..1]
++	date	[1..1]
+	valueDate	[1..1]
++	date	[1..1]
+	bankTransactionCode	[1..1]
++	proprietary	[1..1]
+++	code	[1..1]

+++	issuer	[1..1]
+	entryDetails	[1..1]
++	transactionDetails	[1..1]
+++	references	[0..1]
++++	paymentId	[0..1]
++++	messageIdentification	[0..1]
++++	accountServicerReference	[0..1]
++++	paymentInformationIdentification	[0..1]
++++	instructionIdentification	[0..1]
++++	endToEndIdentification	[0..1]
++++	mandateIdentification	[0..0]/[1..1]
++++	chequeNumber	[0..1]
++++	clearingSystemReference	[0..1]
+++	amountDetails	[0..1]
++++	instructedAmount	[0..1]
+++++	amount	[1..1]
+++++	value	[1..1]
+++++	currency	[1..1]
++++	transactionAmount	[0..0]
+++++	amount	[1..1]
+++++	value	[1..1]
+++++	currency	[1..1]
++++	counterValueAmount	[0..1]
+++++	amount	[1..1]
+++++	value	[1..1]
+++++	currency	[1..1]
+++++	currencyExchange	[0..1]
+++++	sourceCurrency	[1..1]
+++++	targetCurrency	[0..1]
+++++	exchangeRate	[1..1]

++++	proprietaryAmount	[0..0]
+++++	type	[0..0]
+++++	amount	[0..0]
+++++	value	[1..1]
+++++	currency	[1..1]
+++	relatedParties	[1..1]
++++	debtor	[1..1]
+++++	name	[1..1]
+++++	postalAddress	[0..1]
+++++	streetName	[0..1]
+++++	buildingNumber	[0..1]
+++++	postCode	[0..1]
+++++	townName	[0..1]
+++++	country	[0..1]
+++++	addressLine	[0..2]
+++++	identification	[0..1]
+++++	organisationIdentification	[1..1]...either or privateIdentification
+++++	bicOrBei	[0..1]
+++++	other	[0..n]
+++++	identification	[1..1]
+++++	schemeName	[0..1]
+++++	code	[1..1]...either or proprietary
+++++	proprietary	[1..1]...either or code
+++++	issuer	[0..1]
+++++	privateIdentification	[1..1]...either or organisationIdentification
+++++	other	[0..n]
+++++	identification	[1..1]
+++++	schemeName	[0..1]

+++++++	code	[1..1]...either or proprietary
+++++++	proprietary	[1..1]...either or code
+++++++	issuer	[0..1]
++++	debtorAccount	[1..1]
+++++	identification	[1..1]
+++++	iban	[1..1]
+++++	name	[0..1]
++++	ultimateDebtor	[0..1]
+++++	name	[0..1]
+++++	postalAddress	[0..1]
+++++	streetName	[0..1]
+++++	buildingNumber	[0..1]
+++++	postCode	[0..1]
+++++	townName	[0..1]
+++++	country	[0..1]
+++++	addressLine	[0..2]
+++++	identification	[0..1]
+++++	organisationIdentification	[1..1]...either or privateIdentification
+++++	bicOrBei	[0..1]
+++++	other	[0..n]
+++++++	identification	[1..1]
+++++++	schemeName	[0..1]
+++++++	code	[1..1]...either or proprietary
+++++++	proprietary	[1..1]...either or code
+++++++	issuer	[0..1]
+++++	privateIdentification	[1..1]...either or organisationIdentification
+++++	other	[0..n]
+++++++	identification	[1..1]

+++++++	schemeName	[0..1]
+++++++	code	[1..1]...either or proprietary
+++++++	proprietary	[1..1]...either or code
+++++++	issuer	[0..1]
++++	creditor	[1..1]
+++++	name	[1..1]
+++++	postalAddress	[0..1]
+++++	streetName	[0..1]
+++++	buildingNumber	[0..1]
+++++	postCode	[0..1]
+++++	townName	[0..1]
+++++	country	[0..1]
+++++	addressLine	[0..2]...either or individual elements of a structured address
+++++	identification	[0..1]
+++++	organisationIdentification	[1..1]...either or privateIdentification
+++++++	bicOrBei	[0..1]
+++++++	other	[0..n]
+++++++	identification	[1..1]
+++++++	schemeName	[0..1]
+++++++	code	[1..1]...either or proprietary
+++++++	proprietary	[1..1]...either or code
+++++++	issuer	[0..1]
+++++	privateIdentification	[1..1]...either or organisationIdentification
+++++	other	[0..n]
+++++++	identification	[1..1]
+++++++	schemeName	[0..1]
+++++++	code	[1..1]...either or proprietary

+++++++	proprietary	[1..1]...either or code
+++++++	issuer	[0..1]
++++	creditorAccount	[1..1]
+++++	identification	[1..1]
+++++	iban	[1..1]
+++++	name	[0..1]
++++	ultimateCreditor	[0..1]
+++++	name	[0..1]
+++++	postalAddress	[0..1]
+++++	streetName	[0..1]
+++++	buildingNumber	[0..1]
+++++	postCode	[0..1]
+++++	townName	[0..1]
+++++	country	[0..1]
+++++	addressLine	[0..2]
+++++	identification	[0..1]
+++++	organisationIdentification	[1..1] either or privateIdentification
+++++	bicOrBei	[0..1]
+++++	other	[0..n]
+++++	identification	[1..1]
+++++	schemeName	[0..1]
+++++	code	[1..1]...either or proprietary
+++++	proprietary	[1..1]...either or code
+++++	issuer	[0..1]
+++++	privateIdentification	[1..1] either or organisationIdentification
+++++	other	[0..n]
+++++	identification	[1..1]
+++++	schemeName	[0..1]

+++++++	code	[1..1]...either or proprietary
+++++++	proprietary	[1..1]...either or code
+++++++	issuer	[0..1]
+++	relatedAgents	[1..1]
++++	debtorAgent	[1..1]
+++++	financialInstitutionIdentification	[1..1]
+++++	bic	[1..1]
+++++	name	[0..1]
++++	creditorAgent	[1..1]
+++++	financialInstitutionIdentification	[1..1]
+++++	bic	[1..1]
+++++	name	[0..1]
+++	purpose	[0..1]
++++	code	[1..1]...either
++++	proprietary	[1..1]...or
+++	remittanceInformation	[0..1]
++++	unstructured	[0..1]
++++	structured	[0..1]
+++++	creditorReferenceInformation	[0..1]
+++++	reference	[0..1]
+++	additionalTransactionInformation	[0..1]

3.1.5.1.5 Foreign payment – occurrence of elements

LEVEL	MESSAGE ELEMENT	OCCURRENCE
+	entryReference	[0..1]
+	amount	[1..1]
++	value	[1..1]
++	currency	[1..1]
+	creditDebitIndicator	[1..1]

+	reversalIndicator	[0..1]
+	status	[1..1]
+	bookingDate	[1..1]
++	date	[1..1]
+	valueDate	[1..1]
++	date	[1..1]
+	bankTransactionCode	[1..1]
++	proprietary	[1..1]
+++	code	[1..1]
+++	issuer	[1..1]
+	entryDetails	[1..1]
++	transactionDetails	[1..1]
+++	references	[0..1]
++++	paymentId	[0..1]
++++	messageIdentification	[0..1]
++++	accountServicerReference	[0..1]
++++	paymentInformationIdentification	[0..1]
++++	instructionIdentification	[0..1]
++++	endToEndIdentification	[0..1]
++++	chequeNumber	[0..1]
++++	clearingSystemReference	[0..1]
+++	amountDetails	[0..1]
++++	instructedAmount	[0..1]
+++++	amount	[1..1]
+++++	value	[1..1]
+++++	currency	[1..1]
++++	transactionAmount	[0..0]
+++++	amount	[1..1]
+++++	value	[1..1]
+++++	currency	[1..1]

++++	counterValueAmount	[0..1]
+++++	amount	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
+++++	currencyExchange	[0..1]
++++++	sourceCurrency	[1..1]
++++++	targetCurrency	[0..1]
++++++	exchangeRate	[1..1]
++++	proprietaryAmount	[0..0]
+++++	type	[0..0]
+++++	amount	[0..0]
++++++	value	[1..1]
++++++	currency	[1..1]
+++	charges	[0..1]
++++	bearer	[0..1]
+++	relatedParties	[1..1]
++++	debtor	[1..1]
+++++	name	[1..1]
+++++	postalAddress	[0..1]
++++++	streetName	[0..1]
++++++	buildingNumber	[0..1]
++++++	postCode	[0..1]
++++++	townName	[0..1]
++++++	country	[0..1]
++++++	addressLine	[0..2]...either or individual elements of a structured address
++++	debtorAccount	[1..1]
+++++	identification	[1..1]
++++++	iban	[1..1]..either or other identification

+++++	other	[1..1].. either or IBAN
+++++	identification	[1..1]
+++++	name	[0..1]
++++	creditor	[1..1]
+++++	name	[1..1]
+++++	postalAddress	[0..1]
+++++	streetName	[0..1]
+++++	buildingNumber	[0..1]
+++++	postCode	[0..1]
+++++	townName	[0..1]
+++++	country	[0..1]
+++++	addressLine	[0..2]...either or individual elements of a structured address
++++	creditorAccount	[1..1]
+++++	identification	[1..1]
+++++	iban	[1..1].. either or other identification
+++++	other	[1..1].. either or IBAN
+++++	identification	[1..1]
+++++	name	[0..1]
+++	relatedAgents	[1..1]
++++	debtorAgent	[1..1]
+++++	financialInstitutionIdentification	[1..1]
+++++	bic	[0..1]...either
+++++	clearingSystemMemberIdentification	[0..1]...or
+++++	clearingSystemIdentification	[0..1]
+++++	code	[1..1]...either
+++++	proprietary	[1..1]...or
+++++	memberIdentification	[1..1]
+++++	name	[0..1]...or

+++++	postalAddress	[0..1]...or
+++++	streetName	[0..1]
+++++	buildingNumber	[0..1]
+++++	postCode	[0..1]
+++++	townName	[0..1]
+++++	country	[0..1]
+++++	addressLine	[0..2]...either or individual elements of a structured address
+++++	other	[0..1]...or
+++++	identification	[1..1]
++++	creditorAgent	[1..1]
+++++	financialInstitutionIdentification	[1..1]
+++++	bic	[0..1]...either
+++++	clearingSystemMemberIdentification	[0..1]...or
+++++	clearingSystemIdentification	[0..1]
+++++	code	[1..1]...either
+++++	proprietary	[1..1]...or
+++++	memberIdentification	[1..1]
+++++	name	[0..1]...or
+++++	postalAddress	[0..1]...or
+++++	streetName	[0..1]
+++++	buildingNumber	[0..1]
+++++	postCode	[0..1]
+++++	townName	[0..1]
+++++	country	[0..1]
+++++	addressLine	[0..2]...either or individual elements of a structured address
+++++	other	[0..1]...or
+++++	identification	[1..1]

+++	remittanceInformation	[0..1]
++++	unstructured	[0..1]
++++	structured	[0..1]
+++++	creditorReferenceInformation	[0..1]
+++++	reference	[0..1]
+++	additionalTransactionInformation	[0..1]

3.1.5.1.6 Cash – occurrence of elements

LEVEL	MESSAGE ELEMENT	OCCURRENCE
+	entryReference	[0..1]
+	amount	[1..1]
++	value	[1..1]
++	currency	[1..1]
+	creditDebitIndicator	[1..1]
+	reversalIndicator	[0..1]
+	status	[1..1]
+	bookingDate	[1..1]
++	date	[1..1]
+	valueDate	[1..1]
++	date	[1..1]
+	bankTransactionCode	[1..1]
++	proprietary	[1..1]
+++	code	[1..1]
+++	issuer	[1..1]
+	entryDetails	[0..1]
++	transactionDetails	[0..1]
+++	references	[0..1]
++++	messageIdentification	[0..1]
++++	accountServicerReference	[0..1]

++++	paymentInformationIdentification	[0..1]
++++	instructionIdentification	[0..1]
++++	endToEndIdentification	[0..1]
++++	chequeNumber	[0..1]
++++	clearingSystemReference	[0..1]
+++	amountDetails	[0..1]
++++	instructedAmount	[0..1]
+++++	amount	[1..1]
+++++	value	[1..1]
+++++	currency	[1..1]
++++	transactionAmount	[0..0]
+++++	amount	[1..1]
+++++	value	[1..1]
+++++	currency	[1..1]
++++	counterValueAmount	[0..1]
+++++	amount	[1..1]
+++++	value	[1..1]
+++++	currency	[1..1]
+++++	currencyExchange	[0..1]
+++++	sourceCurrency	[1..1]
+++++	targetCurrency	[0..1]
+++++	exchangeRate	[1..1]
++++	proprietaryAmount	[0..0]
+++++	type	[0..0]
+++++	amount	[0..0]
+++++	value	[1..1]
+++++	currency	[1..1]
+++	relatedParties	[0..1]
++++	debtor	[0..1]
+++++	name	[0..1]

++++	debtorAccount	[0..1]
+++++	identification	[1..1]
++++++	iban	[1..1]..either or other identification
++++++	other	[1..1].. either or IBAN
+++++++	identification	[1..1]
++++	creditor	[0..1]
+++++	name	[0..1]
++++	creditorAccount	[0..1]
+++++	identification	[1..1]
++++++	iban	[1..1].. either or other identification
++++++	other	[1..1].. either or IBAN
+++++++	identification	[1..1]
+++	relatedAgents	[0..1]
++++	debtorAgent	[0..1]
+++++	financialInstitutionIdentification	[1..1]
++++++	bic	[1..1]...either or other identification
++++++	name	[0..1]
++++++	other	[1..1]...either or BIC
+++++++	identification	[1..1]
++++	creditorAgent	[0..1]
+++++	financialInstitutionIdentification	[1..1]
++++++	bic	[1..1]...either other identification
++++++	name	[0..1]
++++++	other	[1..1]...either or BIC
+++++++	identification	[1..1]
+++	remittanceInformation	[0..1]
++++	unstructured	[0..1]

++++	structured	[0..1]
+++++	creditorReferenceInformation	[0..1]
++++++	reference	[0..1]
+++	additionalTransactionInformation	[0..1]

3.1.5.1.7 Card transactions – occurrence of elements

LEVEL	MESSAGE ELEMENT	OCCURRENCE
+	entryReference	[0..1]
+	amount	[1..1]
++	value	[1..1]
++	currency	[1..1]
+	creditDebitIndicator	[1..1]
+	reversalIndicator	[0..1]
+	status	[1..1]
+	bookingDate	[1..1]
++	date	[1..1]
+	valueDate	[1..1]
++	date	[1..1]
+	holdExpiration	[0..1]
++	date	[1..1]
+	bankTransactionCode	[1..1]
++	proprietary	[1..1]
+++	code	[1..1]
+++	issuer	[1..1]
+	entryDetails	[0..1]
++	transactionDetails	[0..1]
+++	references	[0..1]

++++	messageIdentification	[0..1]
++++	accountServicerReference	[0..1]
++++	paymentInformationIdentification	[0..1]
++++	instructionIdentification	[0..1]
++++	endToEndIdentification	[0..1]
++++	chequeNumber	[0..1]
++++	clearingSystemReference	[0..1]
+++	amountDetails	[0..1]
++++	instructedAmount	[0..1]
+++++	amount	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
++++	transactionAmount	[0..0]
+++++	amount	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
++++	counterValueAmount	[0..1]
+++++	amount	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
+++++	currencyExchange	[0..1]
++++++	sourceCurrency	[1..1]
++++++	targetCurrency	[0..1]
++++++	exchangeRate	[1..1]
++++	proprietaryAmount	[0..0]
+++++	type	[0..0]
+++++	amount	[0..0]
++++++	value	[1..1]
++++++	currency	[1..1]
+++	relatedParties	[0..1]

++++	debtor	[0..1]
+++++	name	[0..1]
++++	debtorAccount	[0..1]
+++++	identification	[1..1]
+++++	iban	[1..1]..either or other identification
+++++	other	[1..1]..either or IBAN
+++++	identification	[1..1]
++++	creditor	[0..1]
+++++	name	[0..1]
++++	creditorAccount	[0..1]
+++++	identification	[1..1]
+++++	iban	[1..1]..either or other identification
+++++	other	[1..1].. either or IBAN
+++++	identification	[1..1]
+++	relatedAgents	[0..1]
++++	debtorAgent	[0..1]
+++++	financialInstitutionIdentification	[1..1]
+++++	bic	[1..1]...either or other identification
+++++	name	[0..1]
+++++	other	[1..1]...either or BIC
+++++	identification	[1..1]
++++	creditorAgent	[0..1]
+++++	financialInstitutionIdentification	[1..1]
+++++	bic	[1..1]...either or other identification
+++++	name	[0..1]
+++++	other	[1..1]...either or BIC
+++++	identification	[1..1]

+++	remittanceInformation	[0..1]
++++	unstructured	[0..1]
++++	structured	[0..1]
+++++	creditorReferenceInformation	[0..1]
+++++	reference	[0..1]
+++	additionalTransactionInformation	[0..1]

3.1.6 Overview of awaiting transactions (GET /my/accounts/{id}/payments/awaiting){?size,page,sort,order}

Paged list of client's awaiting payments on specific accounts. Each payment contains a unique id usable for URI referencing, e.g. for the payment detail.

Additionally, the debtorAccount contains an API account id to call the account details resource to which the payment order is issued.

Resource characteristics

Mandatory: no

URI: /my/accounts/{id}/payments/awaiting/{?size,page,sort,order}

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Account%20information/-getAccountAwaitingTransactions>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: GET

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: yes

Sorting: yes

Filtering: no

Input character set: no restrictions in relation to chapter 3.1.1

Path parameters of the request:

PARAMETER	TYPE	MANDATORY	PURPOSE
id	Text	Yes	API payment account identifier from the response to a query on the list of accounts.

Query parameters of the request:

PARAMETER	TYPE	MANDATORY	PURPOSE
size	Number	No	Paging. Number of entries per page
page	Number	No	Paging. Required page. + Default: 0
sort	Text	No	A list of fields separated by comma for sorting, arranged according to the meaning
order	Text	No	A list of arrangement methods (ASC, DESC) separated by comma. The order corresponds to the order of fields in the sort parameter.sort parameter.

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is always mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of GET response for calling, please see Chapter 3.1.6.1 RESPONSE MESSAGE ELEMENTS List of client's awaiting payments

Error codes defined for the GET service, List of client's awaiting payments:

HTTP STATUS CODE	ERROR CODE	PURPOSE
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated

401	UNAUTHORISED	Invalid/missing certificate = provider is not authenticated
403	FORBIDDEN	Authentication with an invalid certificate or expired access token, a call that does not match a third-party license.
400	PAGE_NOT_FOUND	Request for a non-existing page
400	PARAMETER_INVALID	The value of the parameter is not valid

3.1.6.1 MESSAGE ELEMENTS Response for GET list of client's awaiting payments

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+	payments	[1..1]	±	Payment collection
++	paymentIdentification	[1..1]	Max35Text	Payment identification
+++	paymentID	[0..1]	Max255Text	Id of initiated payment. The identifier of the payment from the PISP service
+++	instructionIdentification	[1..1]	Max35Text	Instruction identification
+++	transactionIdentification	[1..1]	Max35Text	Transaction identification
+++	endToEndIdentification	[0..1]	Max35Text	End to end identification
++	paymentTypeInformation	[1..1]	PaymentTypeInformation19	Payment type information
+++	instructionPriority	[0..1]	Priority2Code	Instruction priority
+++	serviceLevel	[1..1]	ServiceLevel8CZ	Service level
++++	code	[0..0]	ExternalServiceLevel1Code	Service level code
+++	categoryPurpose	[0..0]	CategoryPurpose1Choice	Category of the purpose of payment
++++	code	[0..0]	ExternalCategoryPurpose1Code	Payment purpose category code
++++	proprietary	[0..0]	Max35Text	Free format for the purpose of the payment

				purpose
++	amount	[1..1]	The same object as to initiate payments chapter ...	Payment amount
++	requestedExecutionDate	[0..1]	ISODate	Requested date of payment
++	debtorAccount	[1..1]	±	Payer account
+++	id	[1..1]	Max35Text	API Payment account identifier
+++	identification	[1..1]	AccountIdentification4Ch oiceCZ	Identification of the payer's account
++++	iban	[1..1]	IBAN2007Identifier	Payer account number in IBAN format. either this parameter or other
++++	other	[1..1]	GenericAccountIdentificat ion1CZ	Payer account number in other/local format. either this parameter or iban
+++++	identification	[1..1]	Max35Text	Payer account number in local BBAN format
+++	currency	[0..1]	CurrencyCode ISO 4217	Payer's account currency
++	creditorAgent	[0..0]	The same object as to initiate payments chapter ...	Bank of reciever
++	creditorAccount	[1..1]	The same object as to initiate payments chapter ...	Payee account
++	remittanceInformation	[0..1]	The same object as to initiate payments chapter ...	Payment information
++	awaitingStatus	[1..1]	FUTURE - payment waiting for execution date. OVERDUE - for example, in case of insufficient	Awaiting status of payment.

			funds at the account OTHER - other kind of payment status	
--	--	--	---	--

3.1.7 List of client's standing orders (GET /my/standingorders{?size,page,sort,order})

Paged list of client's standing orders. Each standing order contains a unique id usable for URI referencing, e.g. for the standing order detail.

It is at the choice of each bank whether to return a list of all the client's standing orders or just those that the client has established via a particular TPP.

Additionally, the debtorAccount contains an API account id to call the account details resource to which the payment order is issued.

Resource characteristics

Mandatory: no

URI: /my/standingorders{?size,page,sort,order}

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Standing%20Orders/-getStandingorders>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: GET

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: yes

Sorting: yes

Filtering: no

Input character set: no restrictions in relation to chapter 3.1.1

Query parameters of the request:

PARAMETER	TYPE	MANDATORY	PURPOSE
size	Number	No	Paging. Number of entries per page

page	Number	No	Paging. Required page. + Default: 0
sort	Text	No	A list of fields separated by comma for sorting, arranged according to the meaning
order	Text	No	A list of arrangement methods (ASC, DESC) separated by comma. The order corresponds to the order of fields in the sort parameter.

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is always mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Response headers](#).

The content of GET response for calling, please see Chapter 3.1.3.1 RESPONSE MESSAGE ELEMENTS List of client standing orders

Error codes defined for the GET service, List of client payment accounts:

<i>HTTP STATUS CODE</i>	<i>ERROR CODE</i>	<i>PURPOSE</i>
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
401	UNAUTHORISED	Invalid/missing certificate = provider is not authenticated
403	FORBIDDEN	Authentication with an invalid certificate or expired access token, a call that does not match a third-party license.
400	PAGE_NOT_FOUND	Request for a non-existing page
400	PARAMETER_INVALID	The value of the parameter is not valid

3.1.7.1 MESSAGE ELEMENTS Response for GET list of standing orders

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+	standingOrders	[1..1]	±	Standing orders collection
++	standingOrderIdentification	[1..1]	Max35Text	Standing order identification
+++	instructionIdentification	[1..1]	Max35Text	Instruction identification
+++	transactionIdentification	[1..1]	Max35Text	Transaction identification
++	amount	[1..1]	STDO - AmountType3CZ ESST - AmountType3CZ XBST - AmountType3Choice	Payment amount
+++	instructedAmount	[1..1]	CurrencyAndAmount	Amount and currency in the instruction
++++	value	[1..1]	Amount	Amount of the transfer
++++	currency	[1..1]	CurrencyCode	Transfer Currency
++	standingOrder	[1..1]	Structure	Structure describing the parameters of the standing orders.
+++	alias	[0..1]	Max250Text	Description resp. the user-defined standing order payment name.
+++	execution	[1..1]	±	Features of execution a standing order for payment.
++++	mode	[1..1]	Max35Text	The execution mode defines when or how standing order will be cancelled, processed the last time. Possible

				<p>values: UNTIL_DATE (standing order is valid until specific date - field lastExecutionDate), UNTIL_CANCELLATION (standing order is valid forever and must be cancelled by client), AFTER_MAX_ITERATION_EXCEEDED (certain count of executions is specified - field maxIterations) or MAX_AMOUNT_EXCEEDED (maximum amount which can be transferred for this order is specified, if next iteration would exceed this amount it is not executed - field maxAmount).</p>
++++	modeDue	[1..1]	Max35Text	<p>The execution due mode defines how the date when order should be executed is specified. Possible values: DUE_DAY_OF_MONTH (specific number of day in the month is defined) or DUE_OR_BEFORE_DAY_OF_MONTH (is defined as a specific day of the month or the previous day if the day falls on a non-banking day) or DUE_OR_NEXT_DAY_OF_MONTH (is defined as a specific day of the month or the next day if the day falls on a non-banking day) or DUE_LAST_DAY_OF_MONTH (order is executed on last day of particular</p>

				month).
++++	interval	[1..1]	Max10Text	Execution interval defines how often order is executed. Possible values: DAILY, WEEKLY, MONTHLY, BI_MONTHLY, QUARTERLY, HALFYEARLY, YEARLY, IRREGULAR.
++++	intervalDue	[1..1]	Number	Value represents order number of the day within particular period when the standing order will be regularly executed. Possible values: 1-7 (for WEEKLY interval), 1-28 for MONTHLY, 1-2 for BI_MONTHLY, 1-3 for QUARTERLY, 1-6 for HALFYEARLY, 1-12 for YEARLY.
++	debtorAccount	[1..1]	±	Payer's account
+++	id	[1..1]	Text	API Payment account identifier
+++	identification	[1..1]	±	Payer account identifier
++++	iban	[1..1]	IBAN2007Identifier	IBAN
+++	currency	[1..1]	CurrencyCode, ISO 4217	Payer account currency

3.1.8 Standing order detail (GET /my/standingorders/{transactionIdentification})

Resource for get detail about the standing order.

It is at the choice of each bank whether to return details of any client's standing order or just those that the client has established through a particular TPP.

Resource characteristics

Mandatory: no

URI: /my/standingorders/{transactionIdentification}

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Standing%20Orders/-getStandingorder>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: GET

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: no

Sorting: no

Filtration: no

Input character set: no restrictions in relation to chapter 3.1.1

Query parameters of the request:

PARAMETER	TYPE	MANDATORY	PURPOSE
transactionIdentification	Max35Text	Yes	Identifier of standing order instruction

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is always mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Response headers](#).

The content of GET response for calling, please see Chapter 3.1.7.1 MESSAGE ELEMENTS Response for GET standing order detail

Error codes defined for the service GET info on entered/initiated standing order:

HTTP STATUS CODE	ERROR CODE	PURPOSE
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
401	UNAUTHORISED	Invalid/missing certificate = provider is not authenticated
501	NOT_IMPLEMENTED	Method not implemented
404	TRANSACTION_MISSING	The requested transaction ID does not exist.

3.1.8.1 MESSAGE ELEMENTS Response for GET standing order detail

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
+	standingOrderIdentification	[1..1]	ALL	±	Standing order identification
++	instructionIdentification	[1..1]	ALL	Max35Text	Instruction identification
++	endToEndIdentification	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	End to end identification
++	transactionIdentification	[0..0]	ALL	Max35Text	Transaction identification
+	paymentTypeInfo	[1..1]	ALL	The same object as to initiate payments chapter ...	
+	amount	[1..1]	ALL	STDO - AmountType3CZ ESST - AmountType3CZ XBST - AmountType3Choice	Payment amount
++	instructedAmount	[1..1]	ALL	±	Amount and currency in the

					instruction
+++	value	[1..1]	ALL	Amount	Amount of the transfer
+++	currency	[1..1]	ALL	CurrencyCode	Transfer Currency
+	requestedExecutionDate	[1..1]	ALL	ISODate	Requested date of payment
+	standingOrder	[1..1]	ALL	±	Structure describing the parameters of the standing orders.
++	alias	[0..1]	ALL	Max250Text	Description resp. the user-defined standing order payment name.
++	execution	[1..1]	ALL	±	Features of execution a standing order for payment.
+++	mode	[1..1]	ALL	Max35Text	The execution mode defines when or how standing order will be cancelled, processed the last time. Possible values: UNTIL_DATE (standing order is valid until specific date - field lastExecutionDate), UNTIL_CANCELLATION (standing order is valid forever and must be cancelled by client), AFTER_MAX_ITERATION_EXCEEDED (certain count of executions is specified - field maxIterations) or MAX_AMOUNT_EXCEEDED (maximum amount which can be transferred for this order is

					specified, if next iteration would exceed this amount - it is not executed - field maxAmount).
+++	modeDue	[1..1]	ALL	Max35Text	The execution due mode defines how the date when order should be executed is specified. Possible values: DUE_DAY_OF_MONTH (specific number of day in the month is defined) or DUE_OR_BEFORE_DAY_OF_MONTH (is defined as a specific day of the month or the previous day if the day falls on a non-banking day) or DUE_OR_NEXT_DAY_OF_MONTH (is defined as a specific day of the month or the next day if the day falls on a non-banking day) or DUE_LAST_DAY_OF_MONTH (order is executed on last day of particular month).
+++	interval	[1..1]	ALL	Max10Text	Execution interval defines how often order is executed. Possible values: DAILY, WEEKLY, MONTHLY, BI_MONTHLY, QUARTERLY, HALFYEARLY, YEARLY, IRREGULAR.
+++	intervalDue	[1..1]	ALL	Number	Value represents order number of the day within particular

					period when the standing order will be regularly executed. Possible values: 1-7 (for WEEKLY interval), 1-28 for MONTHLY, 1-2 for BI_MONTHLY, 1-3 for QUARTERLY, 1-6 for HALFYEARLY, 1-12 for YEARLY.
++	validity	[0..1]	ALL	±	Elements restricting the validity of the standing order.
+++	lastExecutionDate	[0..1]	ALL	ISODate	Date when the last order will be processed. Null value responds to ending by the user.
+++	maxIterations	[0..1]	ALL	Number	Maximum number of iterations - processing of the standing order. Only applicable in combination with executionMode AFTER_MAX_ITERATION_EXCEEDED.
+++	maxAmount	[0..1]	ALL	±	Maximum amount to be transferred using the standing order. Only applicable in combination with executionMode AFTER_MAX_AMOUNT_EXCEEDED.
++++	value	[1..1]	ALL	Amount	Maximum amount of the transfer
++++	currency	[1..1]	ALL	CurrencyCode	Transfer Currency of the maximum amount
++	exceptions	[0..1]	ALL	±	Elements defining

					exceptions to executing a standing order.
+++	stoppages	[0..1]	ALL	Array of Max20Text	List of months where there is no payment. Possible values: JANUARY, FEBRUARY, MARCH, APRIL, MAY, JUNE, JULY, AUGUST, SEPTEMBER, OCTOBER, NOVEMBER, DECEMBER
+++	breaks	[0..1]	ALL	Array of Intervals	Break periods.
++++	validFromDate	[1..1]	ALL	ISODate	Start date of one break period. Standing order will not be processed from this date.
++++	validToDate	[1..1]	ALL	ISODate	End date of one break period. Standing order will not be processed to this date.
+	exchangeRateInformation	[0..0]	ALL	The same object as to initiate payments chapter ...	
+	chargeBearer	[0..0] [0..0] [0..1] [0..1]	TUZEM SEPA EHP NONEHP	ChargeBearerType1 Code	Fee payer
+	chargesAccount	[0..0]	ALL	The same object as to initiate payments chapter ...	
+	ultimateDebtor	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	The same object as to initiate payments chapter ...	

+	debtor	[0..0]	ALL	The same object as to initiate payments chapter ...	
+	debtorAccount	[1..1]	ALL	The same object as to initiate payments chapter ...	
+	intermediaryAgent1	[0..0]	ALL	The same object as to initiate payments chapter ...	
+	creditorAgent	[0..0] [0..1] [1..1] [1..1]	TUZEM SEPA EHP NONEHP	The same object as to initiate payments chapter ...	
+	creditor	[0..0] [1..1] [1..1] [1..1]	TUZEM SEPA EHP NONEHP	The same object as to initiate payments chapter ...	
+	creditorAccount	[1..1]	ALL	The same object as to initiate payments chapter ...	
+	ultimateCreditor	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	The same object as to initiate payments chapter ...	
+	purpose	[0..0]	ALL	The same object as to initiate payments chapter ...	

3.2 API Payment Initiation

3.2.1 How to read API Payment Initiation

The sender of query for API account information must respect permitted characters. Otherwise, the failure to respect them may lead to rejection.

The defined list of enabled characters mainly applies to initiating a payment. For clarification, refer to each resource in the appropriate chapter. In any case, it is necessary to follow the definition in Chapter 1.2.5.

None of these elements may contain separate "/" (slash) at the beginning or end, or two successive slashes in the text. The permitted character set for payment initialisation is based only on the swift character set (that is, exclusively without diacritics - a different character set from the supported characters in CERTIS), i.e. they are the following characters:

```
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
```

Only one query can be sent and processed per call.

All bank account numbers of bank clients in the Czech Republic are defined according to Decree No. 169/2011 Coll. on the rules for the creation of account numbers for making payments. The data element „debtorAccount.identification.iban“ requires an account number in the IBAN format which is defined by the international standard ISO 13616.

Structure of the table of elements:

- **LEVEL** – the character plus „+“ determines the level of element embedding. The basic level is marked „+“ and each another is marked with another character, for instance, the level two is „++“
- **MESSAGE ELEMENT** – element name in the camelCase format
- **OCCURRENCE** – describes the occurrence of fields:
 - [1..1] element is mandatory and only occurs once. In the case of a parent element, at least one embedded element must be filled in. If a mandatory element is embedded, the obligation does not pass to the parent element.
 - [0..1] element is optional and occurs only once.
 - [1..n] element is mandatory and occurs n-times. If the value "n" is not numerically defined, the number of repetitions is unlimited.

- [0..n] element is not mandatory and occurs n-times. If the value "n" is not numerically defined, the number of repetitions is unlimited.
- [0..0] element is not mandatory and in the specified type (request, attribute or element) does not have to occur, i.e. such element does not have to be implemented by a bank at all.
- **PAYMENT TYPE** – defines for which message types the current element is relevant
- **FORMAT TYPE** – defines the data format. It may be determined by the ISO 20022 standard valid for the item type PAIN.001 or another standard specifying the data structure (e.g., changes, dates, etc). Some items have a specific required format resulting from the payment system environment in the Czech Republic.
- **PRESENTATION** – contains a general role description

The format type defined as „±“ means the parent element which is then subdivided into other elements.

Conditions for element presence:

- When there is no OR but the parent element has more than 2 levels of elements, elements may be filled in cumulatively
- Parent element after [1..1] – at least one embedded element must be filled in

3.2.2 List of API Payment Initiation resources

Specification of API Payment Initiation contains a description of resources for the payment initiation and the Balance Check of the payer (client) at a specific financial institution.

Overview of resources:

- POST query for balance check
- POST new single payment (Payment Initiation)
- GET status of entered/initiated single payment
- GET information on entered/initiated single payment
- DELETE deletion of the entered and not authorised single payment
- POST generation of authorization ID for single payment
- Single payment authorisation
 - GET Step I. payment authorisation detail
 - POST Step II. payment authorisation initiation - specific for each bank
 - PUT Step III. payment authorisation finalization - specific for each bank

- POST new standing order
- GET standing order detail
- GET standing order status
- DELETE unauthorised standing order
- PUT standing order change
- Standing order authorization
 - GET Step I. standing order authorization detail
 - POST Step II. standing order authorization initiation - specific for each bank
 - PUT Step III. Standing order authorization finalization - specific for each bank
- Batch payment - bulk authorization
- Batch payment delete
- Batch payment status
- Batch payment authorization
 - Batch payment authorization step I (GET)
 - Batch payment authorization step II (POST)
 - Batch payment authorization step III (PUT)
- Get list of PSU authorizations
- List of client's awaiting payments
- Payment initialization with one SCA
- Overview of unsecured loans initiated via TPP
- GET Loan offers
- POST Loan application
- Get Loan application status
- POST Initiation of drawing a loan
- Loan application cancellation
- Domestic direct debit consent - instruction initiation
- Domestic direct debit Consent detail
- List of client's domestic direct debits Consents
- Change of existing Domestic direct debit Consent
- Domestic direct debit Consent delete

- Direct debit authorization Step I
- Direct debit authorization Step II
- Direct debit authorization Step III

3.2.3 Query for Balance Check (POST /my/payments/balanceCheck)

This is the resource for sending a request for balance check in a particular payer's payment account. This resource is authorized. Access to information must be granted by the client outside the interaction of this API before the resource is used.

Resource characteristics

Mandatory: yes

URI: /my/payments/balanceCheck

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Payment%20Initiation/-postBalanceCheck>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: POST

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: no

Sorting: no

Filtering: no

Input character set: limitation is based on definitions from chapter 3.2.1

Query parameters of the request: **not defined**

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is always mandatory for this resource within PIS calls

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Response headers](#).

The content of POST request and response for calling, please see Chapter 3.2.3.1 MESSAGE ELEMENTS Query for balance check

Error codes defined for the POST service Query for balance check

<i>HTTP STATUS CODE</i>	<i>ERROR CODE</i>	<i>PURPOSE</i>
401	UNAUTHORISED	Missing certificate.
403	FORBIDDEN	Calling of the method which does not correspond to the licence, or invalid certificate.
400	FIELD_MISSING	Missing mandatory field in the request.
400	FIELD_INVALID	FIELD value is not valid.
400	AC02	[InvalidDebtorAccountNumber] – invalid account identifier in the request content.
400	AC09	[InvalidAccountCurrency] – invalid currency of the required account.
400	AC12	[InvalidAccountType] - account type does not match allowed account types (e.g., a non-paying account).
403	AG01	[TransactionForbidden] – absent consent to access to balance check at the account.
400	AM11	[InvalidTransactionCurrency] – the request contains a currency not trade/not supported.
400	AM12	[InvalidAmount] – wrong amount. For instance, too low or high amount or wrong number format according to the number of decimal places according to the ISO 4217.
400	FF01	[Invalid File Format] – invalid JSON format or other technical problem with the query processing.
400, 50x	NARR	Narrative – a general reason for rejecting the payment, with an addition of error-related information.

400	RF01	[NotUniqueTransactionReference] – not unique request identifier.
400	RR10	[InvalidCharacterSet] – invalid character set in the request.

3.2.3.1 MESSAGE ELEMENTS Query for Balance Check

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+	exchangeIdentification	[1..1]	Max18Text	Clear query identification
+	card	[0..1]	±	Transaction card
++	cardholderName	[0..1]	Max45Text	Card holder name
++	maskedPan	[1..1]	Max30Text	Masked card number
+	debtorAccount	[1..1]	±	Payer account
++	identification	[1..1]	±	Payer account identification
+++	iban	[1..1]	IBAN2007Identifier	IBAN
++	currency	[0..1]	CurrencyCode, ISO 4217	Payer account currency
+	authenticationMethod	[0..1]	CodeSet	Client verification method
+	merchant	[0..1]	±	Merchant executing the transaction
++	identification	[1..1]	Max35Text	Merchant identification
++	type	[0..1]	Code	Merchant type
++	shortName	[1..1]	Max35Text	Merchant name

++	commonName	[1..1]	Max70Text	Merchant name as stated in the payment receipt
++	address	[0..1]	Max140Text	Merchant address
++	countryCode	[0..1]	CountryCode, ISO 3166 (2 alphanumeric characters code version)	Merchant country
++	merchantCategoryCode	[1..1]	Min3Max4Text, ISO 18245	Merchant code following the transaction type
+	transactionDetails	[1..1]	±	Transaction details
++	currency	[1..1]	CurrencyCode, ISO 4217	Balance query currency
++	totalAmount	[1..1]	Amount	Balance query amount

3.2.3.2 MESSAGE ELEMENTS Response for Balance Check

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+	responseIdentification	[1..1]	Number (integre)	Unique identification of response to query for Balance Check (from ASPSP).
+	exchangeIdentification	[1..1]	Max18Text	Repeated identification of a payment transaction (query for Balance Check) from the issuer of the card to which the request for Balance Check linked to the account.
+	response	[1..1]	Code set	Result code of query for Balance

				Check.
--	--	--	--	--------

Return codes for the parameter „response“ – Code set:

CODE	DESCRIPTION
APPR	Enough funds on this account
DECL	Unsufficient funds on this account

3.2.4 New payment - payment initiation (POST /my/payments)

Resource for entering a new payment.

Resource characteristics

Mandatory: yes

URI: /my/payments

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Payment%20Initiation/-postNewPayment>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: POST

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: no

Sorting: no

Filtration: no

Input character set: limitation is based on definitions from chapter 3.2.1

Query parameters of the request: **not defined**

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is always mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of POST request and response for calling, please see Chapter 3.2.4.1 MESSAGE ELEMENTS New payment - payment initiation

Error codes defined for the service POST Query for Payment Initiation

<i>HTTP STATUS CODE</i>	<i>ERROR CODE</i>	<i>PURPOSE</i>
401	UNAUTHORISED	Missing certificate.
403	FORBIDDEN	Calling of the method which does not correspond to the licence, or invalid certificate.
400	FIELD_MISSING	Missing mandatory field in the request.
400	FIELD_INVALID	FIELD value is not valid.
400	AC02	[InvalidDebtorAccountNumber] – invalid account identifier in the request content.
400	AC03	[InvalidCreditorAccountNumber] - recipient account number is closed, blocked, credit card are disabled for the account type, or the recipient's account number is in invalid format (note: only valid for in-house payments).
400	AC10	[InvalidDebtorAccountCurrency] – the specified payer account currency does not correspond to the client's account currency for the account number held in the bank (the account currency is optional, in the case of multicurrency accounts, the account currency must be specified - Raiffeisenbank).
400	AC12	[InvalidAccountType] - account type does not match allowed account types (e.g., a non-paying account).

403	AG01	[TransactionForbidden] – absent consent to access to Balance Check at the account.
400	AM05	[Duplication] - this is a duplicate. Universal code for a duplicate CISP query (validated by, for example, CSOB) or a duplicate payment via GIS (non-unique payment reference).
400	AM11	[InvalidTransactionCurrency] – the request contains a currency not trade/not supported.
400	AM12	[InvalidAmount] – wrong amount. For instance, too low or high amount or wrong number format according to the number of decimal places according to the ISO 4217.
400	FF01	[Invalid File Format] – invalid JSON format or other technical problem with the query processing.
400	BE19	[InvalidChargeBearerCode] - invalid fee type for the given transaction type.
400	DT01	[InvalidDate] - non-existent date value or format. Maturity in the future or in the past. Transactions on a non-business day of a bank (unless the bank receives such orders).
400, 50x	NARR	Narrative – a general reason for rejecting the payment, with an addition of error-related information.
400	RC07	[InvalidCreditorBICIdentifier] – invalid recipient SWIFT / BIC code.
400	RC10	[InvalidCreditorClearingSystemMemberIdentifier] - invalid recipient bank code identification.
400	RR03	[MissingCreditorNameOrAddress] - the field is missing the required data about the recipient's name or address (or its part). If the data is in bad format, the error code FIELD_INVALID is used.
400	RR10	[InvalidCharacterSet] – invalid character set in the request.
400	NO_PART	<p>Unattainable Payment Method - The payer or payee bank does not provide the requested priority of the instruction (Instant Payment).</p> <p>In this case, the bank can extend the error response by the "parameters" element and define a set of possible payment priorities. TPP resp. the PSU may be able to choose another type of priority.</p> <p>...</p> <pre>"parameters": { "ALLOWED_PRIORITY_TYPES" : ["NORM", "HIGH"]</pre>

		} ...
400	REC_SEND	[RECEIVER_SAME_AS_SENDER] – Payer's account is the same as the payee's account (Applies to all types of payments)

An overview of error codes that may occur in the case of instant payments.

HTTP STATUS CODE	ERROR CODE	PURPOSE
400	CNOR	[Creditor bank is not registered] - The beneficiary's bank is not a participant in the Czech instant payments scheme.
400	DNOR	[Debtor bank is not registered] - The payer's bank is not a participant in the Czech instant payments scheme.
400	AB08	[Offline Creditor Agent] – The beneficiary's bank is currently unavailable due to planned and unplanned outages of the Czech instant payments scheme.
400	AM02	[Not Allowed Amount] - Exceeded limit for single instant payment.
400	AG09	[Payment Not Received] - The beneficiary's bank rejected to accept instant payment to the beneficiary's account.
400	FORB	[Transaction Forbidden] - The payer's bank doesn't allow instant payment from the payer's account.
400	AB07	[Offline Agent] - The ČNB is not currently available for the Czech instant payments.
400	AB05	[Timeout Creditor Agent] - Transaction declined due to timeout.
400	OFFD	[Offline Debtor Agent] - The payer's bank is currently unavailable due to planned and unplanned outages of the Czech instant payments scheme

3.2.4.1 MESSAGE ELEMENTS New payment – payment initiation

Payment types considered

PAYMENT CODE	SERVICELLEVELCODE	DESCRIPTION
TUZEM	DMCT	Domestic payment

SEPA	ESCT	SEPA payment
EHP	EXCT	Foreign payment within EEA
NONEHP	NXCT	Foreign payment outside EEA

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
+	paymentIdentification	[1..1]	ALL	Max35Text	Payment identification
++	instructionIdentification	[1..1]	ALL	Max35Text	Instruction identification
++	endToEndIdentification	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	End to end identification
++	transactionIdentification	[0..0]	ALL	Max35Text	Transaction identification
+	paymentTypeInfoInformation	[0..1]	ALL	PaymentTypeInfoInformation19	Payment type information
++	instructionPriority	[0..1]	ALL	Priority2Code	Instruction priority
++	serviceLevel	[0..0]	ALL	ServiceLevel8CZ	Service level
+++	code	[0..0]	ALL	ExternalServiceLevel1Code	Service level code
++	categoryPurpose	[0..0]	ALL	CategoryPurpose1Choice	Category of the purpose of payment
+++	code	[0..0]	ALL	ExternalCategoryPurpose1Code	Payment purpose category code
+++	proprietary	[0..0]	ALL	Max35Text	Free format for the purpose of the payment purpose
+	amount	[1..1]	ALL	TUZEM - AmountType3CZ SEPA -	Payment amount

				AmountType3CZ EHP - AmountType3Choice NONEHP - AmountType3Choice	
++	instructedAmount	[1..1]	ALL	CurrencyAndAmount	Amount and currency in the instruction
+++	value	[1..1]	ALL	Amount	Amount of the transfer
+++	currency	[1..1]	ALL	CurrencyCode	Transfer Currency
++	equivalentAmount	[0..0]	ALL	CurrencyAndAmount	Equivalent amount and currency
+++	value	[0..0]	ALL	Amount	Equivalent transaction amount
+++	currency	[0..0]	ALL	CurrencyCode	Currency equivalent to transaction amount
+	requestedExecutionDate	[0..1]	ALL	ISODate	Requested date of payment
+	exchangeRateInformation	[0..0]	ALL	ExchangeRateInformation1	Contractual course
++	exchangeRate	[0..0]	ALL	BaseOneRate	Agreed exchange rate
++	rateType	[0..0]	ALL	ExchangeRateType1Code	Type of agreed exchange rate
++	contractIdentification	[0..0]	ALL	Max35Text	Identify the use of the agreed exchange rate
+	chargeBearer	[0..0] [0..0] [0..1] [0..1]	TUZEM SEPA EHP NONEHP	ChargeBearerType1Code	Fee payer
+	chargesAccount	[0..0]	ALL	CashAccount16CZ	Account for fees
++	identification	[0..0]	ALL	AccountIdentificati	Identification of

				on4ChoiceCZ	account number for fees
+++	Iban	[0..0]	ALL	IBAN2007Identifier	Account number for charges in IBAN format
++	currency	[0..0]	ALL	CurrencyCode ISO 4217	Account currency for fees
+	ultimateDebtor	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	PartyIdentification 32CZ1	Original payer
++	Name	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max70Text	The name of the original payer
++	postalAddress	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	PostalAddress6CZ	Postal address of the original payer
+++	streetName	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max70Text	Street of original payer
+++	buildingNumber	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max16Text	The original payer's house number
+++	postCode	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max16Text	The postal code of the original payer
+++	townName	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	City of the original payer
+++	country	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	CountryCode, ISO3166 (2 alphanumeric characters code version)	Country of the original payer

+++	addressLine	[0..0] [0..2] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max70Text	Unstructured entry of the original payer's address
++	identification	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Party6Choice	Identification of the original payer
+++	organisationIdentification	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	OrganisationIdentification4CZ	Unambiguous identification of the original payer as organization / legal entity. either organizationalIdentification or privateIdentification
++++	bicOrBei	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	BICIdentifier	Identification of the original payer as organization / legal entity in the form of BIC or BEI code.
++++	other	[0..0] [0..n] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	GenericOrganisationIdentification1	Other identification of the original payer as organization / legal entity.
+++++	identification	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	Other identification of the original payer as organization / legal entity in unstructured form.
+++++	schemeName	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	OrganisationIdentificationSchemeName1CZ	Document type for identifying the original payer as an organization / legal entity.
+++++	code	[0..0] [1..1]...either or proprietary [0..0]	TUZEM SEPA EHP NONEHP	Code	Document type for the identification of the original payer as as an organization / legal entity in the form of code according to the ISO

		[0..0]			code list.
+++++	proprietary	[0..0] [1..1]...e ither or code [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	Document type for identifying the original payer as an organization / legal entity in free text format.
+++++	issuer	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	Document exhibitor for identification the original payer as organization / legal entity.
+++	privateIdentification	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	PersonIdentification5CZ	Unambiguous identification of the original payer as a natural person. either organizationalIdentification or privateIdentification
++++	other	[0..0] [0..n] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	GenericPersonIdentification1	Other identification of the original payer as a natural person in unstructured form.
+++++	identification	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	Other identification of the original payer as a natural person in unstructured form.
+++++	schemeName	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	PersonIdentificationSchemeName1Choice	Document type to identify the original payer as a natural person.
+++++	code	[0..0] [1..1]...e ither or proprietary [0..0]	TUZEM SEPA EHP NONEHP	Code	Document type for the identification of the original payer as a physical person in the form of code according to the ISO

		[0..0]			code list.
+++++	proprietary	[0..0] [1..1]...e ither or code [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	Document type for identifying the original payer as a physical person in free text format.
+++++	issuer	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	Document exhibitor for identification the original payer as a natural person.
+	debtor	[0..0]	ALL	PartyIdentification 32CZ2	Payer.
++	name	[0..0]	ALL	Max70Text	Payer name
++	postalAddress	[0..0]	ALL	PostalAddress6CZ	Postal address of the payer
+++	streetName	[0..0]	ALL	Max70Text	The street name used for the postal address of the payer.
+++	buildingNumber	[0..0]	ALL	Max16Text	The reference number used for the postal address of the payer.
+++	postCode	[0..0]	ALL	Max16Text	The postal code used for the postal address of the payer.
+++	townName	[0..0]	ALL	Max35Text	The city name used for the postal address of the payer.
+++	country	[0..0]	ALL	CountryCode, ISO3166 (2 alphanumeric characters code version)	The country name used for the postal address of the payer.
+++	addressLine	[0..0]	ALL	Max70Text	Unstructured payer

					postal address record.
+	debtorAccount	[1..1]	ALL	CashAccount16CZ	Payer's account
++	identification	[1..1]	ALL	AccountIdentification4ChoiceCZ	Identification of the payer's account
+++	iban	[1..1]	ALL	IBAN2007Identifier	Payer account number in IBAN format
+++	other	[0..0]	ALL	GenericAccountIdentification1CZ	Another payer account number format
++++	identification	[0..0]	ALL	Max34Text	Payer account number in local BBAN format
++	currency	[0..1]	ALL	CurrencyCode ISO 4217	Payer's account currency
+	intermediaryAgent1	[0..0]	ALL	BranchAndFinancialInstitutionIdentification4CZ	Intermediate 1
++	financialInstitutionIdentification	[0..0]	ALL	FinancialInstitutionIdentification7CZ	Identification of the financial institution
+++	bic	[0..0]	ALL	BICIdentifier	BIC / SWIFT bank code
+++	clearingSystemMemberIdentification	[0..0]	ALL	ClearingSystemMemberIdentification2	Identify a participant in a clearing system
++++	clearingSystemIdentification	[0..0]	ALL	ClearingSystemIdentification2Choice	Identification of the clearing system
+++++	code	[0..0]	ALL	ClearingSystemIdentification1Code	Code
+++++	proprietary	[0..0]	ALL	Max35Text	Free format
++++	memberIdentification	[0..0]	ALL	Max35Text	Subscriber Clearing Code
+++	name	[0..0]	ALL	Max70Text	Name
+++	postalAddress	[0..0]	ALL	PostalAddress6CZ	Postal address
++++	streetName	[0..0]	ALL	Max70Text	Street

++++	buildingNumber	[0..0]	ALL	Max16Text	Building number
++++	postCode	[0..0]	ALL	Max16Text	Zip code
++++	townName	[0..0]	ALL	Max35Text	City
++++	country	[0..0]	ALL	CountryCode, ISO3166 (2 alphanumeric characters code version)	Country code
++++	addressLine	[0..0]	ALL	Max70Text	Unstructured address entry
+++	other	[0..0]	ALL	GenericFinancialIdentification1CZ	Other bank identification
++++	identification	[0..0]	ALL	Max35Text	Local bank code
+	creditorAgent	[0..0] [0..1] [1..1] [1..1]	TUZEM SEPA EHP NONEHP	BranchAndFinancialInstitutionIdentification4CZ	Bank of receiver
++	financialInstitutionIdentification	[0..0] [0..1] [1..1] [1..1]	TUZEM SEPA EHP NONEHP	FinancialInstitutionIdentification7CZ	Identification of the financial institution
+++	bic	[0..0] [1..1] [1..1] [0..1]either or one of the other variants of the beneficiary's bank identification	TUZEM SEPA EHP NONEHP	BICIdentifier	BIC / SWIFT bank code

+++	clearingSystemMemberIdentification	[0..0] [0..0] [0..0] [0..1]... either or any of the other variants of the beneficiary's bank identification	TUZEM SEPA EHP NONEHP	ClearingSystemMemberIdentification2	Identifikace účastníka clearingového systému
++++	clearingSystemIdentification	[0..0] [0..0] [0..0] [0..1]	TUZEM SEPA EHP NONEHP	ClearingSystemIdentification2Choice	Identify a participant in a clearing system
+++++	code	[0..0] [0..0] [0..0] [1..1]... either code or proprietary	TUZEM SEPA EHP NONEHP	ExternalClearingSystemIdentification1Code	Code
+++++	proprietary	[0..0] [0..0] [0..0] [1..1]... either code or proprietary	TUZEM SEPA EHP NONEHP	Max35Text	Free format
++++	memberIdentification	[0..0] [0..0] [0..0] [1..1]	TUZEM SEPA EHP NONEHP	Max35Text	Subscriber Clearing Code
+++	name	[0..0] [0..0] [0..0] [0..1]... either	TUZEM SEPA EHP NONEHP	Max105Text	Name

		or any of the other variants of the beneficiary's bank identification			
+++	postalAddress	[0..0] [0..0] [0..0] [0..1] ... either or any of the other variants of the beneficiary's bank identification	TUZEM SEPA EHP NONEHP	PostalAddress6CZ	Postal address
++++	streetName	[0..0] [0..0] [0..0] [0..1]	TUZEM SEPA EHP NONEHP	Max70Text	Street
++++	buildingNumber	[0..0] [0..0] [0..0] [0..1]	TUZEM SEPA EHP NONEHP	Max16Text	Building number
++++	postCode	[0..0] [0..0] [0..0] [0..1]	TUZEM SEPA EHP NONEHP	Max16Text	ZIP code
++++	townName	[0..0] [0..0] [0..0] [0..1]	TUZEM SEPA EHP NONEHP	Max35Text	City

++++	country	[0..0] [0..0] [0..0] [0..1]	TUZEM SEPA EHP NONEHP	CountryCode, ISO3166 (2 alphanumeric characters code version)	Country code
++++	addressLine	[0..0] [0..0] [0..0] [0..2]	TUZEM SEPA EHP NONEHP	Max70Text	Unstructured address entry
+++	other	[0..0] [0..0] [0..0] [0..1]... either or any of the other variants of the benefici ary's bank identific ation	TUZEM SEPA EHP NONEHP	GenericFinancialIde ntification1C	Other bank identification
++++	identification	[0..0] [0..0] [0..0] [0..1]	TUZEM SEPA EHP NONEHP	Max35Text	Local bank code
+	creditor	[0..0] [1..1] [1..1] [1..1]	TUZEM SEPA EHP NONEHP	PartyIdentification 32CZ2	Payee
++	name	[0..0] [1..1] [1..1] [1..1]	TUZEM SEPA EHP NONEHP	Max70Text	Payee name
++	postalAddress	[0..0] [0..1] [0..1] [1..1]	TUZEM SEPA EHP NONEHP	PostalAddress6CZ	Postal address

+++	streetName	[0..0] [0..1] [0..1] [0..1]	TUZEM SEPA EHP NONEHP	Max70Text	Street
+++	buildingNumber	[0..0] [0..1] [0..1] [0..1]	TUZEM SEPA EHP NONEHP	Max16Text	Building number
+++	postCode	[0..0] [0..1] [0..1] [0..1]	TUZEM SEPA EHP NONEHP	Max16Text	ZIP code
+++	townName	[0..0] [0..1] [0..1] [0..1]	TUZEM SEPA EHP NONEHP	Max35Text	City
+++	country	[0..0] [0..1] [0..1] [0..1]	TUZEM SEPA EHP NONEHP	CountryCode, ISO3166 (2 alphanumeric characters code version)	Country
+++	addressLine	[0..0] [0..2] [0..2] [0..2]	TUZEM SEPA EHP NONEHP	Max70Text	Unstructured address entry
+	creditorAccount	[1..1]	ALL	CashAccount16CZ	Payee account
++	identification	[1..1]	ALL	AccountIdentificati on4ChoiceCZ	Payee Account Identification
+++	iban	[1..1]	ALL	IBAN2007Identifier	Account number in IBAN format
+++	other	[0..0] [0..0] [1..1] [1..1]	TUZEM SEPA EHP NONEHP	GenericAccountIde ntification1CZ	Another account number format
++++	identification	[0..0] [0..0] [1..1] [1..1]	TUZEM SEPA EHP NONEHP	Max34Text	Account number in local BBAN format

++	currency	[0..1] [0..0] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	CurrencyCode ISO4217	Payee account currency
+	ultimateCreditor	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	PartyIdentification 32CZ1	The final payee
++	name	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max70Text	Name of the final payee
++	postalAddress	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	PostalAddress6CZ	Postal address of the final payee
+++	streetName	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max70Text	Street
+++	buildingNumber	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max16Text	Building number
+++	postCode	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max16Text	ZIP code
+++	townName	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	City
+++	country	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	CountryCode, ISO3166 (2 alphanumeric characters code version)	Country
+++	addressLine	[0..0] [0..2] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max70Text	Unstructured address entry

++	identification	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Party6Choice	Identification of the final payee
+++	organisationIdentification	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	OrganisationIdentification4CZ	Unambiguous identification of the final payee as an organization / legal entity. either organisationalIdentification or privateIdentification
++++	bicOrBei	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	BICIdentifier	Identification of the final payee as an organization / legal entity in the form of BIC or BEI code.
++++	other	[0..0] [0..n] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	GenericOrganisationIdentification1	Other identification of the final payee as organization / legal entity.
+++++	identification	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	Other identification of parent payee as organization / legal entity in unstructured form.
+++++	schemeName	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	OrganisationIdentificationSchemeName1CZ	Document type for identifying the final payee as an organization / legal entity.
+++++	code	[0..0] [1..1]...either or proprietary [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Code	Document type for the identification of the final payee as an organization / legal entity in the form of code according to the ISO code list.

+++++	proprietary	[0..0] [1..1]...e ither or code [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	Document type for the identification of the final payee as an organization / legal entity in free text format.
+++++	issuer	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	Document exhibitor for the final payee identification as an organization / legal entity.
+++	privateIdentification	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	PersonIdentification5CZ	Unambiguous identification of the final payee as a natural person. either organizationalIdentification or privateIdentification
++++	other	[0..0] [0..n] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	GenericPersonIdentification1	Other identification of the final payee as a natural person in unstructured form.
+++++	identification	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	Other identification of the final payee as a natural person in unstructured form.
+++++	schemeName	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	PersonIdentificationSchemeName1Choice	Document type for identifying the final payee as a natural person.
+++++	code	[0..0] [1..1]...e ither or proprietary [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Code	Document type for the identification of the final payee as a physical person in the form of code according to the ISO code list.
+++++	proprietary	[0..0] [1..1]...e ither or code [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	Document type for identifying the final payee as a physical person in free text

		[0..0]			format.
+++++	issuer	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	Document exhibitor for the final payee identification as a natural person.
+	purpose	[0..0]	ALL	Purpose2Choice	Purpose of payment
++	code	[0..0]	ALL	ExternalPurpose1Code	Purpose of payment code
++	proprietary	[0..0]	ALL	Max35Text	Free form of payment purpose
+	instructionForNextAgent	[0..0]	ALL	Instruction code	Instructions for the next bank
+	remittanceInformation	[0..1]	ALL	RemittanceInformation5CZ	Payment information
++	unstructured	[0..1]	ALL	Max140Text	Unstructured message for the payee
++	structured	[0..1] [0..0] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	StructuredRemittanceInformation7CZ	Structured message for payee - variable, specific, and constant symbol
+++	creditorReferenceInformation	[0..1] [0..0] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	CreditorReferenceInformation2CZ	Payee reference information
++++	reference	[0..3] [0..0] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	CreditorReferenceInformation2CZ	Value of VS, SS, CS
++	additionalRemittanceInformation	[0..1]	ALL	Max140Text	Additional free text information for payer of up to 140

					characters.
--	--	--	--	--	-------------

3.2.4.2 MESSAGE ELEMENTS Response for new payment – payment initiation

The table contains only the elements that appear only in the response message.

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+	paymentIdentification	[1..1]	±	Payment identification
++	transactionIdentification	[1..1]	Max35Text	Identifier of established transaction
+	paymentTypeInfoInformation	[1..1]	±	Payment type information
++	serviceLevel	[1..1]	±	Service level
+++	code	[1..1]	ServiceLevelCode	Service level code
+	signInfo	[1..1]	±	Status information and id of unauthorized transactions
++	state	[1..1]	StateCode	Transaction authorization status
++	signId	[0..1]	Text	Identifier of the authorization process of a particular transaction.
+	instructionStatus	[1..1]	StatusCode	Status of entered payment
+	statusChangeInfo	[0..1]	Text	Specifies the change which was made to the instruction. Used with instructionStatus "ACWC".

Values of serviceLevel.code - The type of entered payment

CODE	DESCRIPTION
DMCT	[DoMestic Credit Transfer] Domestic payment
ESCT	[SEPA Credit Transfer] – SEPA payment
XBCT	[Cross-Border Credit Transfer] – Cross border payment

EXCT	[EEA X-border Credit Transfers] – Cross border payment within the EEP
NXCT	[Non-EEA X-border Credit Transfers] – Cross border payment outside the EEP

Status codes of payment – StatusCode

HTTP STATUS CODE	STATUS CODE	PURPOSE
200	ACTC	[AcceptedTechnicalValidation] - Authentication and syntactical and semantical validation are successful
200	RJCT	[Rejected] - Payment initiation or individual transaction included in the payment initiation has been rejected
200	ACWC	[AcceptedWithChange] - Instruction is accepted but a change will be made, such as date or remittance not sent

3.2.5 Status of entered/initiated payment (GET /payments/{paymentId}/status)

A resource to display the payment status. It is an entered payment which has not been authorised by the client yet or has already been authorised, and PISP requires its status (GET).

The resource returns the information only for transactions entered through a specific provider. Information on the provider is taken over from the certificate or licence information.

The user authorization of this resource is optional. Primarily, only a valid certificate of the provider is demanded

Resource characteristics

Mandatory: yes

URI: /payments/{paymentId}/status

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Payment%20Initiation/-getPaymentStatus>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: GET

Authorization: request **does not require** the authorization of user/client as part of the API calling

Use certificate: request **requires** the use of the third-party qualified certificate

Paging: no

Sorting: no

Filtering: no

Input character set: no restrictions in relation to chapter 3.2.1

Query parameters of the request: **not defined**

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is optional for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of GET response calling, please see Chapter 3.2.5.1 MESSAGE ELEMENTS Status of entered/initiated payment

Error codes defined for the service GET status of entered/initiated payment:

<i>HTTP STATUS CODE</i>	<i>ERROR CODE</i>	<i>PURPOSE</i>
401	UNAUTHORISED	Invalid/missing certificate = provider is not authenticated
404	TRANSACTION_MISSING	The requested transaction ID does not exist.

3.2.5.1 MESSAGE ELEMENTS Status of entered/initiated payment

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
+	instructionStatus	[1..1]	PISP ALL	StatusCode	Status of entered payment
+	statusChangeInfo	[0..1]	PISP ALL	Text	Specifies the change which was made to the instruction. Used with instructionStatus "ACWC".

Status codes of payments – StatusCode

HTTP STATUS CODE	STATUSCODE	PURPOSE
200	ACTC	[AcceptedTechnicalValidation] - Authentication and syntactical and semantical validation are successful
200	RJCT	[Rejected] - Payment initiation or individual transaction included in the payment initiation has been rejected
200	ACSP	[AcceptedSettlementInProgress] - All preceding checks such as technical validation and customer profile were successful and therefore the payment initiation has been accepted for execution
200	ACSC	[AcceptedSettlementCompleted] - Settlement on the debtor's account has been completed. Usage: this can be used by the first agent to report to the debtor that the transaction has been completed. Warning: this status is provided for transaction status reasons, not for financial information. It can only be used after bilateral agreement
200	ACWC	[AcceptedWithChange] - Instruction is accepted but a change will be made, such as date or remittance not sent

3.2.6 Info on entered/initiated payment (GET /my/payments/{paymentId})

A resource to display the information on the entered payment. It is payment which is received for authorisation, but has not been authorised by the client yet. The resource only works with transactions entered **through a specific provider**.

The resource to find out the transaction detail. Information on the provider is taken from the certificate or information on licence.

Resource characteristics

Mandatory: no

URI: /my/payments/{paymentId}

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Payment%20Initiation/-getPaymentInfo>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: GET

Authorization: request **requires** the authorization of user/client as part of the API calling

Use certificate: request **requires** the use of the third-party qualified certificate

Paging: no

Sorting: no

Filtering: no

Input character set: no restrictions in relation to chapter 3.2.1

Query parameters of the request: **not defined**

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is always mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of GET response for calling, please see Chapter 3.2.6.1 MESSAGE ELEMENTS Info on entered/initiated payment

Error codes defined for the service GET info on entered/initiated payment:

HTTP STATUS CODE	ERROR CODE	PURPOSE
401	UNAUTHORISED	Invalid/missing certificate = provider not authorised
501	NOT_IMPLEMENTED	Method not implemented
404	TRANSACTION_MISSING	The requested transaction ID does not exist.

3.2.6.1 MESSAGE ELEMENTS Info on entered/initiated payment

The result of the message is an overview of entered or already initiated payment. Therefore, the list of elements corresponds to the elements from resource New payment, see 3.2.4.1 MESSAGE ELEMENTS New payment - payment initiation.

3.2.7 Delete an entered payment (DELETE /my/payments/{paymentId})

A resource for deleting **unauthorised and authorised payments**.

Deleting of an unauthorised payment is not conditioned by the transaction authorisation as it is not a payment received by the bank.

Deleting of an authorised payment must be followed by the authorisation process which is identical with the new payment authorisation process.

Resource characteristics

Mandatory: no

URI: /my/payments/{paymentId}

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Payment%20Initiation/deleteUnauthorisedPayment>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: DELETE

Authorization: request **requires** the authorization of user/client as part of the API calling

Use certificate: request **requires** the use of the third-party qualified certificate

Paging: no

Sorting: no

Filtering: no

Input character set: no restrictions in relation to chapter 3.2.1

Query parameters of the request: **not defined**

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is always mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Response headers](#).

Error codes defined for the service DELETE an entered payment:

<i>HTTP STATUS CODE</i>	<i>ERROR CODE</i>	<i>PURPOSE</i>
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
403	FORBIDDEN	Invalid/missing certificate = provider is not authenticated
501	NOT_IMPLEMENTED	Method not implemented
404	TRANSACTION_MISSING	The requested transaction ID does not exist.

3.2.7.1 MESSAGE ELEMENTS Delete an entered payment

In case of an unauthorised payment, no content is returned in the response if the deleting was successful.

In case of an authorised payment, the response contains a body (specified here below) with the payment identification and also signId, which is necessary to complete the deleting of the authorised payment process.

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
+	paymentIdentification	[1..1]	ALL	±	Payment identification
++	instructionIdentification	[1..1]	ALL	Max35Text	Instruction identification
++	endToEndIdentification	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	End to end identification
++	transactionIdentification	[1..1]	ALL	Max35Text	Transaction identification
+	signInfo	[1..1]	ALL	±	Information on the current instruction authorization
++	state	[1..1]	PISP ALL	Text	Status of the current instruction authorization in a format supported by bank
++	signId	[1..1]	PISP ALL	Text	Unique identifier for the current instruction authorization

3.2.8 Generating the authorization ID (POST /my/payments/{paymentId}/sign)

Generating the authorization ID for payment before the initiation of the authorization process. The resource is implemented by banks that do not want to generate authorization ID for payment in the response when entering a new (unauthorized) payment through the resource POST /my/payments.

Each bank may define and describe in its documentation its own authorization scenarios. The CODE of authorization methods for these scenarios is used in the response of this resource.

The response to this resource is an authorization ID generated for the specific payment returned in the element signInfo.

Part of the response is also the set of scenarios that may contain one or more authorisation scenarios. If more scenarios are returned, it is up to the client to select what scenario to choose. The power of verifying each scenario should correspond to the power of other scenarios, or the scenarios are mutually replaceable.

Each scenario is defined as the sequence of codes of particular authorization methods. The code of authorization method is defined by the bank and should be described in its own API documentation outside the ČOBS standard.

Resource characteristics

Mandatory: no

URI: /my/payments/{paymentId}/sign

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Payment%20Initiation/-postGeneratingAuthorizationId>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP method: POST

Authorization: request **requires** the authorization of user/client as part of the API calling

Use certificate: request **requires** the use of the third-party qualified certificate

Paging: no

Sorting: no

Filtering: no

Input character set: no restrictions in relation to chapter 3.2.1

Query parameters of the request: **not defined**

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is always mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of POST request and response for calling, please see Chapter 3.2.8.1 MESSAGE ELEMENTS Generating the authorization ID

Error codes defined for the POST service generating the authorization ID:

<i>HTTP STATUS CODE</i>	<i>ERROR CODE</i>	<i>PURPOSE</i>
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
403	FORBIDDEN	Invalid/missing certificate = provider is not authenticated
501	NOT_IMPLEMENTED	Method not implemented
404	TRANSACTION_MISSING	The requested transaction ID does not exist.

3.2.8.1 MESSAGE ELEMENTS Generating the authorization ID

<i>LEVEL</i>	<i>MESSAGE ELEMENT</i>	<i>OCCURRENCE</i>	<i>PAYMENT TYPE</i>	<i>FORMAT TYPE</i>	<i>PRESENTATION</i>
+	scenarios	[1..1]	PISP ALL	±	A set of possible authorization scenarios
+	signInfo	[1..1]	PISP ALL	±	Information on instruction authorization
++	state	[1..1]	PISP ALL	Text	Status of transaction authorization in a format supported by bank
++	signId	[1..1]	PISP ALL	Text	Unique identifier for current transaction

					authorization
--	--	--	--	--	---------------

3.2.9 Step I. Payment authorization detail (GET /my/payments/{paymentId}/sign/{signId})

An optional resource to identify the status and scenarios of payment authorization.

Part of the response is also the set of scenarios that may contain one or more authorisation scenarios. If more scenarios are returned, it is up to the client to select what scenario to choose. The power of verifying each scenario should correspond to the power of other scenarios, or the scenarios are mutually replaceable.

Each scenario is defined as the sequence of codes of particular authorization methods. Each bank may define and describe in its documentation its own authorization scenarios. CODE of authorization methods of these scenarios is used in the response of this resource.

Resource characteristics

Mandatory: no

URI: /my/payments/{paymentId}/sign/{signId}

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Payment%20Initiation/-getPaymentAuthorizationDetail>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: GET

Authorization: request **requires** the authorization of user/client as part of the API calling

Use certificate: request **requires** the use of the third-party qualified certificate

Paging: no

Sorting: no

Filtering: no

Input character set: no restrictions in relation to chapter 3.2.1

Query parameters of the request: **not defined**

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is always mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of GET requests and response for calling, please see Chapter 3.2.9.1 MESSAGE ELEMENTS Step I. Payment authorization detail

Error codes defined for the service GET step I. payment authorization detail:

<i>HTTP STATUS CODE</i>	<i>ERROR CODE</i>	<i>PURPOSE</i>
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
403	FORBIDDEN	Invalid/missing certificate = provider is not authenticated
501	NOT_IMPLEMENTED	Method not implemented
404	ID_NOT_FOUND	The required id does not exist
400	AUTH_LIMIT_EXCEEDED	The resource may not be authorized with this method

3.2.9.1 MESSAGE ELEMENTS Step I. Payment authorization detail

<i>LEVEL</i>	<i>MESSAGE ELEMENT</i>	<i>OCCURRENCE</i>	<i>PAYMENT TYPE</i>	<i>FORMAT TYPE</i>	<i>PRESENTATION</i>
+	scenarios	[1..1]	PISP ALL	±	A set of possible authorization scenarios
+	signInfo	[1..1]	PISP ALL	±	Information on

					instruction authorization
++	state	[1..1]	PISP ALL	Text	Status of transaction authorization in a format supported by the bank
++	signId	[1..1]	PISP ALL	Text	Unique identifier for current transaction authorization

3.2.10 Step II. Payment authorization initiation - specific for each bank (POST /my/payments/{paymentId}/sign/{signId})

This resource is designed to **start a specific authorization method** from the selected scenario.

The input is the JSON object containing the required type of authorization method - **CODE** and all elements specific for that step.

The output of this resource is an overview of values required for completing the authorization.

E.g., for CODE corresponding to the federated authorization, the response will be URL and parameters for redirecting to the federated authorization page.

And, for instance, for CODE corresponding to authorization through the OTP code sent by SMS, the response will only be the confirmation of sending the code. The sending as such is initiated by the bank.

Resource characteristics

Mandatory: yes

URI: /my/payments/{paymentId}/sign/{signId}

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Payment%20Initiation/-postPaymentAuthorizationInitiation>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: POST

Authorization: request **requires** the authorization of user/client as part of the API calling

Use certificate: request **requires** the use of the third-party qualified certificate

Paging: no

Sorting: no

Filtering: no

Input character set: no restrictions in relation to chapter 3.2.1

Query parameters of the request: **not defined**

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is always mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of POST request and response for calling, please see chapter 3.2.10.1 MESSAGE ELEMENTS Step II. Payment authorization initiation - specific for each bank

Error codes defined for the POST service Step II. Payment authorization initiation:

<i>HTTP STATUS CODE</i>	<i>ERROR CODE</i>	<i>PURPOSE</i>
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
403	FORBIDDEN	Invalid/missing certificate = provider is not authenticated
404	ID_NOT_FOUND	The required id does not exist
400	AUTH_LIMIT_EXCEEDED	The resource may not be authorized with this method
400	INVALID_AUTHORI	The value of redirect uri is not valid (not registered)

	ZATION_REDIRECT_URI	
--	---------------------	--

3.2.10.1 MESSAGE ELEMENTS Step II. Payment authorization initiation – specific for each bank

Request parameters:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
+	authorizationType	[1..1]	PISP ALL	Text	Code of required authorization (from authorization scenarios)
+	redirectUrl	[0..1]	PISP ALL	Text	<p>URL link or package of federated authorization call back address.</p> <p>This address is used by the federated bank authorization to redirect back to the TPP application after authorization is complete.</p> <p>One of the registered authorization_redirect_urls must be used</p>

Response parameters:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
+	authorizationType	[1..1]	PISP ALL	±	Code of required <i>authorization (from authorization scenarios)</i>
+	Href	[0..1]	PISP ALL	±	Reference to call federated authorization
++	url	[1..1]	PISP ALL	Text	URL link or package of federated authorization
++	Id	[0..1]	PISP ALL	Text	Possible id for calling federated authorization
+	Method	[0..1]	PISP ALL	Text	Method to use href link and federated authorization.
+	formData	[0..1]	PISP ALL	±	Optional element. For the method POST federated authorization (authorizationType=USERAGENT_REDIRECT)

					CT) element Contains data for sending in redirecting to federated authorization.
++	SAMLRequest	[0..1]	PISP ALL	Text	Optional parameter. For the method POST federated authorization (authorizationType=USERAGENT_REDIRECT) element Contains data of SAML request
++	relayState	[0..1]	PISP ALL	Text	Optional parameter. For the method POST federated authorization (authorizationType=USERAGENT_REDIRECT) element Contains relayState for returnable value.
+	signInfo	[1..1]	PISP ALL	±	Information on instruction author
++	State	[1..1]	PISP ALL	Text	Status of transaction authorization in a format supported by the bank
++	signId	[1..1]	PISP ALL	Text	Unique identifier for current transaction authorization

3.2.11 Step III. Payment authorization initiation – specific for each bank (PUT /my/payments/{paymentId}/sign/{signId})

An optional resource designed for the finalization of the authorization process. The resource may be purely indicative or returns information on whether the authorization process was completed and with what result. E.g., in the case of federated authorization. Or what input, e.g. for entering the received OTP for the SMS method. Then the output is information on the authorization and also OTP verification.

Resource characteristics

Mandatory: no

URI: /my/payments/{paymentId}/sign/{signId}

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Payment%20Initiation/-putPaymentAuthorizationInitiation>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: PUT

Authorization: request **requires** the authorization of user/client as part of the API calling

Use certificate: request **requires** the use of the third-party qualified certificate

Paging: no

Sorting: no

Filtering: no

Input character set: no restrictions in relation to chapter 3.2.1

Query parameters of the request: **not defined**

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is always mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of PUT request and response for calling, please see Chapter 3.2.11.1 MESSAGE ELEMENTS Step III. Payment authorization finalization - specific for each bank

Error codes defined for the PUT service Step III. Payment authorization initiation:

<i>HTTP STATUS CODE</i>	<i>ERROR CODE</i>	<i>PURPOSE</i>
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
403	FORBIDDEN	Invalid/missing certificate = provider is not authenticated
501	NOT_IMPLEMENTED	Method not implemented
404	ID_NOT_FOUND	The required id does not exist
400	AUTH_LIMIT_EXCEEDED	The resource may not be authorized with this method

3.2.11.1 MESSAGE ELEMENTS Step III. Payment authorization finalization - specific for each bank

Request parameters

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
+	authorizationType	[1..1]	PISP ALL	Text	Code of required authorization (from authorization scenarios)
+	oneTimePassword	[0..1]	PISP ALL	Max35Text	One-time password received e.g. in SMS.

Response parameters

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
+	state	[1..1]	PISP ALL	Text	Status of transaction authorization in a format supported by the bank
+	instructionStatus	[0..1]	PISP ALL	StatusCode	Status of entered payment
+	pollInterval	[0..1]	PISP ALL	Number	Value in milliseconds how often I can ask for authorization status.

Status codes of payment – StatusCode

HTTP STATUS CODE	STATUS CODE	PURPOSE
200	ACTC	[AcceptedTechnicalValidation] - Authentication and syntactical and semantical validation are successful
200	RJCT	[Rejected] - Payment initiation or individual transaction included in the payment initiation has been rejected

200	ACWC	[AcceptedWithChange] - Instruction is accepted but a change will be made, such as date or remittance not sent
-----	------	---

3.2.12 Standing orders - recurring payments

Overview of resources for the establishment, authorization and administration of standing orders.

3.2.13 New standing order - instruction initiation (POST /my/standingorders)

Resource for entering a new standingorder.

Resource characteristics

Mandatory: yes

URI: /my/standingorders

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Standing%20Orders/-postNewStandingOrder>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: POST

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: no

Sorting: no

Filtration: no

Input character set: limitation is based on definitions from chapter 3.2.1

Query parameters of the request: **not defined**

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is always mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of POST request and response for calling, please see Chapters 3.2.13.1 MESSAGE ELEMENTS New standing order initiation and 3.2.13.2 MESSAGE ELEMENTS Response for new standing order initiation

Error codes defined for the service POST Query for Payment Initiation

<i>HTTP STATUS CODE</i>	<i>ERROR CODE</i>	<i>PURPOSE</i>
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
401	UNAUTHORISED	Invalid/missing certificate = provider is not authenticated
403	FORBIDDEN	Calling of the method which does not correspond to the licence, or invalid certificate.
400	FIELD_MISSING	Missing mandatory field in the request.
400	FIELD_INVALID	FIELD value is not valid.
400	AC02	[InvalidDebtorAccountNumber] – invalid account identifier in the request content.
400	AC03	[InvalidCreditorAccountNumber] - recipient account number is closed, blocked, credit card are disabled for the account type, or the recipient's account number is in invalid format (note: only valid for in-house payments).
400	AC10	[InvalidDebtorAccountCurrency] – the specified payer account currency does not correspond to the client's account currency for the account number held in the bank (the account currency is optional, in the case of multicurrency accounts, the account currency must be specified - Raiffeisenbank).
400	AC12	[InvalidAccountType] - account type does not match allowed account types (e.g., a non-paying account).
403	AG01	[TransactionForbidden] – absent consent to access to Balance Check at the account.
400	AM05	[Duplication] - this is a duplicate. Universal code for a duplicate CISP query (validated by, for example, CSOB) or a duplicate payment via GIS (non-unique payment reference).

400	AM11	[InvalidTransactionCurrency] – the request contains a currency not trade/not supported.
400	AM12	[InvalidAmount] – wrong amount. For instance, too low or high amount or wrong number format according to the number of decimal places according to the ISO 4217.
400	FF01	[Invalid File Format] – invalid JSON format or other technical problem with the query processing.
400	BE19	[InvalidChargeBearerCode] - invalid fee type for the given transaction type.
400	DT01	[InvalidDate] - non-existent date value or format. Maturity in the future or in the past. Transactions on a non-business day of a bank (unless the bank receives such orders).
400, 50x	NARR	Narrative – a general reason for rejecting the payment, with an addition of error-related information.
400	RC07	[InvalidCreditorBICIdentifier] – invalid recipient SWIFT / BIC code.
400	RC10	[InvalidCreditorClearingSystemMemberIdentifier] - invalid recipient bank code identification.
400	RR03	[MissingCreditorNameOrAddress] - the field is missing the required data about the recipient's name or address (or its part). If the data is in bad format, the error code FIELD_INVALID is used.
400	RR10	[InvalidCharacterSet] – invalid character set in the request.
400	REC_SEND	[RECEIVER_SAME_AS_SENDER] – Payer's account is the same as the payee's account (Applies to all types of payments)

3.2.13.1 MESSAGE ELEMENTS New standing order initiation

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
+	standingOrderIdentification	[1..1]	ALL	Max35Text	Standing order identification
++	instructionIdentification	[1..1]	ALL	Max35Text	Instruction identification

++	endToEndIdentification	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	End to end identification
++	transactionIdentification	[0..0]	ALL	Max35Text	Transaction identification
+	paymentTypeInformation	[0..1]	ALL	The same object as to initiate payments chapter ...	
+	amount	[1..1]	ALL	STDO - AmountType3CZ ESST - AmountType3CZ XBST - AmountType3Choice	Payment amount
++	instructedAmount	[1..1]	ALL	CurrencyAndAmount	Amount and currency in the instruction
+++	value	[1..1]	ALL	Amount	Amount of the transfer
+++	currency	[1..1]	ALL	CurrencyCode	Transfer Currency
+	requestedExecutionDate	[1..1]	ALL	ISODate	Requested date of payment
+	standingOrder	[1..1]	ALL	Structure	Structure describing the parameters of the standing orders.
++	alias	[0..1]	ALL	Max250Text	Description resp. the user-defined standing order payment name.
++	execution	[1..1]	ALL		Features of execution a standing order for payment.
+++	mode	[1..1]	ALL	Max35Text	The execution mode defines when or how standing order will be cancelled, processed the last

					<p>time. Possible values: UNTIL_DATE (standing order is valid until specific date - field lastExecutionDate), UNTIL_CANCELLATION (standing order is valid forever and must be cancelled by client), AFTER_MAX_ITERATION_EXCEEDED (certain count of executions is specified - field maxIterations) or MAX_AMOUNT_EXCEEDED (maximum amount which can be transferred for this order is specified, if next iteration would exceed this amount it is not executed - field maxAmount).</p>
+++	modeDue	[1..1]	ALL	Max35Text	<p>The execution due mode defines how the date when order should be executed is specified. Possible values: DUE_DAY_OF_MONTH (specific number of day in the month is defined) or DUE_OR_BEFORE_DAY_OF_MONTH (is defined as a specific day of the month or the previous day if the day falls on a non-banking day) or DUE_OR_NEXT_DAY_OF_MONTH (is defined as a specific</p>

					day of the month or the next day if the day falls on a non-banking day) or DUE_LAST_DAY_OF_MONTH (order is executed on last day of particular month).
+++	interval	[1..1]	ALL	Max10Text	Execution interval defines how often order is executed. Possible values: DAILY, WEEKLY, MONTHLY, BI_MONTHLY, QUARTERLY, HALFYEARLY, YEARLY, IRREGULAR.
+++	intervalDue	[1..1]	ALL	Number	Value represents order number of the day within particular period when the standing order will be regularly executed. Possible values: 1-7 (for WEEKLY interval), 1-28 for MONTHLY, 1-2 for BI_MONTHLY, 1-3 for QUARTERLY, 1-6 for HALFYEARLY, 1-12 for YEARLY.
++	validity	[0..1]	ALL		Elements restricting the validity of the standing order.
+++	lastExecutionDate	[0..1]	ALL	ISODate	Date when the last order will be processed. Null value responds to ending by the user.
+++	maxIterations	[0..1]	ALL	Number	Maximum number of iterations - processing of the standing order. Only

					applicable in combination with executionMode AFTER_MAX_ITERATION_EXCEEDED.
+++	maxAmount	[0..1]	ALL	Amount	Maximum amount to be transferred using the standing order. Only applicable in combination with executionMode AFTER_MAX_AMOUNT_EXCEEDED.
++++	value	[1..1]	ALL	Amount	Maximum amount of the transfer
++++	currency	[1..1]	ALL	CurrencyCode	Transfer Currency of the maximum amount
++	exceptions	[0..1]	ALL		Elements defining exceptions to executing a standing order.
+++	stoppages	[0..1]	ALL	Array of Max20Text	List of months where there is no payment. Possible values: JANUARY, FEBRUARY, MARCH, APRIL, MAY, JUNE, JULY, AUGUST, SEPTEMBER, OCTOBER, NOVEMBER, DECEMBER
+++	breaks	[0..1]	ALL	Array of Intervals	Break periods.
++++	validFromDate	[1..1]	ALL	ISODate	Start date of one break period. Standing order will not be processed from this date.
++++	validToDate	[1..1]	ALL	ISODate	End date of one break period. Standing order will

					not be processed to this date.
+	exchangeRateInformation	[0..0]	ALL	The same object as to initiate payments chapter ...	
+	chargeBearer	[0..0] [0..0] [0..1] [0..1]	TUZEM SEPA EHP NONEHP	ChargeBearerType1 Code	Fee payer
+	chargesAccount	[0..0]	ALL	The same object as to initiate payments chapter ...	
+	ultimateDebtor	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	The same object as to initiate payments chapter ...	
+	debtor	[0..0]	ALL	The same object as to initiate payments chapter ...	
+	debtorAccount	[1..1]	ALL	The same object as to initiate payments chapter ...	
+	intermediaryAgent1	[0..0]	ALL	The same object as to initiate payments chapter ...	
+	creditorAgent	[0..0] [0..1] [1..1] [1..1]	TUZEM SEPA EHP NONEHP	The same object as to initiate payments chapter ...	
+	creditor	[0..0] [1..1] [1..1] [1..1]	TUZEM SEPA EHP NONEHP	The same object as to initiate payments chapter ...	
+	creditorAccount	[1..1]	ALL	The same object as to initiate payments chapter ...	

				...	
+	ultimateCreditor	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	The same object as to initiate payments chapter ...	
+	purpose	[0..0]	ALL	The same object as to initiate payments chapter ...	
+	instructionForNextAgent	[0..0]	ALL	Instruction code	Instructions for the next bank
+	remittanceInformation	[0..1]	ALL	The same object as to initiate payments chapter ...	

3.2.13.2 MESSAGE ELEMENTS Response for new standing order initiation

The table contains only the elements that appear only in the response message.

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+	standingOrderIdentification	[1..1]	ALL	Max35Text
++	transactionIdentification	[1..1]	Max35Text	Identifier of established standing order instruction
+	paymentTypeInfoInformation	[1..1]	ALL	
++	serviceLevel	[1..1]	±	Service placement (within type of payment)
+++	code	[1..1]	Text	Type of entered payment
+	signInfo	[1..1]	±	Status information and id of unauthorized transactions
++	state	[1..1]	StateCode	Transaction authorization

				status
++	signId	[1..1]	Text	Identifier of the authorization process of a particular transaction.
+	instructionStatus	[1..1]	StatusCode	Status of entered standing order
+	statusChangeInfo	[0..1]	Text	Specifies the change which was made to the instruction. Used with instructionStatus "ACWC".

Values of serviceLevel.code - The type of entered payment

CODE	DESCRIPTION
DMCT	[DoMestic Credit Transfer] Domestic payment
ESCT	[SEPA Credit Transfer] – SEPA payment
XBCT	[Cross-Border Credit Transfer] – Cross border payment
EXCT	[EEA X-border Credit Transfers] – Cross border payment within the EEP
NXCT	[Non-EEA X-border Credit Transfers] – Cross border payment outside the EEP

Status codes of standing order – StatusCode

HTTP STATUS CODE	STATUS CODE	PURPOSE
200	ACTC	[AcceptedTechnicalValidation] - Authentication and syntactical and semantical validation are successful
200	RJCT	[Rejected] - Payment initiation or individual transaction included in the payment initiation has been rejected
200	ACWC	[AcceptedWithChange] - Instruction is accepted but a change will be made, such as date or remittance not sent

3.2.14 Standing order detail (GET /my/standingorders/{transactionIdentification})

Resource for get detail about the standing order

Resource characteristics

Mandatory: no

URI: /my/standingorders/{transactionIdentification}

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Standing%20Orders/-getStandingorder>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: GET

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: no

Sorting: no

Filtration: no

Input character set: limitation is based on definitions from chapter 3.2.1

Path parameters of the request:

PARAMETER	TYPE	MANDATORY	PURPOSE
transactionIdentification	Max35Text	Yes	Identifier of established standing order instruction

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is always mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of GET response for calling, please see Chapter 3.2.14.1 MESSAGE ELEMENTS Response for GET standing order detail

Error codes defined for the service GET info on entered/initiated standing order:

<i>HTTP STATUS CODE</i>	<i>ERROR CODE</i>	<i>PURPOSE</i>
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
401	UNAUTHORISED	Invalid/missing certificate = provider is not authenticated
501	NOT_IMPLEMENTED	Method not implemented
404	TRANSACTION_MISSING	The requested transaction ID does not exist.

3.2.14.1 MESSAGE ELEMENTS Response for GET standing order detail

<i>LEVEL</i>	<i>MESSAGE ELEMENT</i>	<i>OCCURRENCE</i>	<i>PAYMENT TYPE</i>	<i>FORMAT TYPE</i>	<i>PRESENTATION</i>
+	standingOrderIdentification	[1..1]	ALL	Max35Text	Standing order identification
++	instructionIdentification	[1..1]	ALL	Max35Text	Instruction identification
++	endToEndIdentification	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	End to end identification
++	transactionIdentification	[0..0]	ALL	Max35Text	Transaction identification
+	paymentTypeInfoInformation	[1..1]	ALL	The same object as to initiate payments chapter ...	

+	amount	[1..1]	ALL	STDO - AmountType3CZ ESST - AmountType3CZ XBST - AmountType3Choice	Payment amount
++	instructedAmount	[1..1]	ALL	CurrencyAndAmount	Amount and currency in the instruction
+++	value	[1..1]	ALL	Amount	Amount of the transfer
+++	currency	[1..1]	ALL	CurrencyCode	Transfer Currency
+	requestedExecutionDate	[1..1]	ALL	ISODate	Requested date of payment
+	standingOrder	[1..1]	ALL	Structure	Structure describing the parameters of the standing orders.
++	alias	[0..1]	ALL	Max250Text	Description resp. the user-defined standing order payment name.
++	execution	[1..1]	ALL		Features of execution a standing order for payment.
+++	mode	[1..1]	ALL	Max35Text	The execution mode defines when or how standing order will be cancelled, processed the last time. Possible values: UNTIL_DATE (standing order is valid until specific date - field lastExecutionDate), UNTIL_CANCELLATION (standing order is valid forever and must be cancelled by client), AFTER_MAX_ITERAT

					ION_EXCEEDED (certain count of executions is specified - field maxIterations) or MAX_AMOUNT_EXCEEDED (maximum amount which can be transferred for this order is specified, if next iteration would exceed this amount it is not executed - field maxAmount).
+++	modeDue	[1..1]	ALL	Max35Text	The execution due mode defines how the date when order should be executed is specified. Possible values: DUE_DAY_OF_MONTH (specific number of day in the month is defined) or DUE_OR_BEFORE_DAY_OF_MONTH (is defined as a specific day of the month or the previous day if the day falls on a non-banking day) or DUE_OR_NEXT_DAY_OF_MONTH (is defined as a specific day of the month or the next day if the day falls on a non-banking day) or DUE_LAST_DAY_OF_MONTH (order is executed on last day of particular month).
+++	interval	[1..1]	ALL	Max10Text	Execution interval defines how often order is executed.

					Possible values: DAILY, WEEKLY, MONTHLY, BI_MONTHLY, QUARTERLY, HALFYEARLY, YEARLY, IRREGULAR.
+++	intervalDue	[1..1]	ALL	Number	Value represents order number of the day within particular period when the standing order will be regularly executed. Possible values: 1-7 (for WEEKLY interval), 1-28 for MONTHLY, 1-2 for BI_MONTHLY, 1-3 for QUARTERLY, 1-6 for HALFYEARLY, 1-12 for YEARLY.
++	validity	[0..1]	ALL		Elements restricting the validity of the standing order.
+++	lastExecutionDate	[0..1]	ALL	ISODate	Date when the last order will be processed. Null value responds to ending by the user.
+++	maxIterations	[0..1]	ALL	Number	Maximum number of iterations - processing of the standing order. Only applicable in combination with executionMode AFTER_MAX_ITERATION_EXCEEDED.
+++	maxAmount	[0..1]	ALL	Amount	Maximum amount to be transferred using the standing order. Only applicable in combination with

					executionMode AFTER_MAX_AMOUNT_EXCEEDED.
++++	value	[1..1]	ALL	Amount	Maximum amount of the transfer
++++	currency	[1..1]	ALL	CurrencyCode	Transfer Currency of the maximum amount
++	exceptions	[0..1]	ALL		Elements defining exceptions to executing a standing order.
+++	stoppages	[0..1]	ALL	Array of Max20Text	List of months where there is no payment. Possible values: JANUARY, FEBRUARY, MARCH, APRIL, MAY, JUNE, JULY, AUGUST, SEPTEMBER, OCTOBER, NOVEMBER, DECEMBER
+++	breaks	[0..1]	ALL	Array of Intervals	Break periods.
++++	validFromDate	[1..1]	ALL	ISODate	Start date of one break period. Standing order will not be processed from this date.
++++	validToDate	[1..1]	ALL	ISODate	End date of one break period. Standing order will not be processed to this date.
+	exchangeRateInformation	[0..0]	ALL	The same object as to initiate payments chapter ...	
+	chargeBearer	[0..0] [0..0] [0..1]	TUZEM SEPA EHP	ChargeBearerType1 Code	Fee payer

		[0..1]	NONEHP		
+	chargesAccount	[0..0]	ALL	The same object as to initiate payments chapter ...	
+	ultimateDebtor	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	The same object as to initiate payments chapter ...	
+	debtor	[0..0]	ALL	The same object as to initiate payments chapter ...	
+	debtorAccount	[1..1]	ALL	The same object as to initiate payments chapter ...	
+	intermediaryAgent1	[0..0]	ALL	The same object as to initiate payments chapter ...	
+	creditorAgent	[0..0] [0..1] [1..1] [1..1]	TUZEM SEPA EHP NONEHP	The same object as to initiate payments chapter ...	
+	creditor	[0..0] [1..1] [1..1] [1..1]	TUZEM SEPA EHP NONEHP	The same object as to initiate payments chapter ...	
+	creditorAccount	[1..1]	ALL	The same object as to initiate payments chapter ...	
+	ultimateCreditor	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	The same object as to initiate payments chapter ...	
+	purpose	[0..0]	ALL	The same object as to initiate payments chapter ...	

+	instructionForNextAgent	[0..0]	ALL	Instruction code	Instructions for the next bank
+	remittanceInformation	[0..1]	ALL	The same object as to initiate payments chapter ...	
+	signInfo	[1..1]	ALL	±	Status information and id of unauthorized transactions
++	state	[1..1]	ALL	StateCode	Transaction authorization status
++	signId	[0..1]	ALL	Text	Identifier of the authorization process of a particular transaction.
+	instructionStatus	[1..1]	ALL	Status Code set	Transaction status identifier
+	statusChangeInfo	[0..1]	ALL	Text	Specifies the change which was made to the instruction. Used with instructionStatus "ACWC".

Status codes of standing order – StatusCode

HTTP STATUS CODE	STATUS CODE	PURPOSE
200	ACTC	[AcceptedTechnicalValidation] - Authentication and syntactical and semantical validation are successful
200	RJCT	[Rejected] - Payment initiation or individual transaction included in the payment initiation has been rejected
200	ACWC	[AcceptedWithChange] - Instruction is accepted but a change will be made, such as date or remittance not sent
200	CUST	[RequestedByCustomer] - Cancellation requested by the Debtor
200	ACSC	[AcceptedSettlementCompleted] – Standing order on the debtor's account has been completed

3.2.15 Standing order status (GET */my/standingorders/{transactionIdentification}/status*)

Resource for getting status of the standing order.

Resource characteristics

Mandatory: yes

URI: */my/standingorders/{transactionIdentification}/status*

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Standing%20Orders/-getStandingorderStatus>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: GET

Authorization: request **requires no** authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: no

Sorting: no

Filtration: no

Input character set: limitation is based on definitions from chapter 3.2.1

Query parameters of the request: **not defined**

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is optional for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of GET for request and response calling, please see Chapter 3.2.15.1 MESSAGE ELEMENTS Status of entered/initiated standing order

Error codes defined for the service GET status of entered/initiated standing order:

<i>HTTP STATUS CODE</i>	<i>ERROR CODE</i>	<i>PURPOSE</i>
401	UNAUTHORISED	Invalid/missing certificate = provider is not authenticated
404	TRANSACTION_MISSING	The requested transaction ID does not exist.

3.2.15.1 MESSAGE ELEMENTS Status of entered/initiated standing order

<i>LEVEL</i>	<i>MESSAGE ELEMENT</i>	<i>OCCURRENCE</i>	<i>FORMAT TYPE</i>	<i>PRESENTATION</i>
+	instructionStatus	[1..1]	StatusCode	Status of entered standing order
+	realizedPayments	[0..1] conditional if it is an authorized standing order	Array	Array of transaction identifiers
++	transactionIdentification	[1..1]	Max35Text	Identifier of established transaction
++	date	[0..1]	ISODate/ISODateTime	Due date/payment foreign currency in the format ISODate, , or ISODateTime, i.e. YYYY-MM-DD, or YYYY-MM-DDThh:mm:ss.STZD, depending on the transaction type and method how the bank presents data (and time) of due date/payment foreign currency. Mainly for card or cash transactions it is posted as ISODateTime.

Status codes of standing order – StatusCode

HTTP STATUS CODE	STATUSCODE	PURPOSE
200	ACTC	[AcceptedTechnicalValidation] - Authentication and syntactical and semantical validation are successful
200	RJCT	[Rejected] - Payment initiation or individual transaction included in the payment initiation has been rejected
200	ACSP	[AcceptedSettlementInProgress] - All preceding checks such as technical validation and customer profile were successful and therefore the payment initiation has been accepted for execution
200	CUST	[RequestedByCustomer] - Cancellation requested by the Debtor
200	ACSC	[AcceptedSettlementCompleted] – Standing order on the debtor's account has been completed

3.2.16 Standing order delete (DELETE /my/standingorders/{transactionIdentification})

Resource to delete **unauthorised and authorised standing orders**.

Deleting of an authorised standing order must be followed by the authorisation process which is identical with the new standing order authorisation process.

Resource characteristics

Mandatory: no

URI: /my/standingorders/{transactionIdentification}

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Standing%20Orders/-deleteStandingorder>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: DELETE

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: no

Sorting: no

Filtration: no

Input character set: limitation is based on definitions from chapter 3.2.1

Query parameters of the request: **not defined**

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

Error codes defined for the service DELETE the entered unauthorised standing order:

<i>HTTP STATUS CODE</i>	<i>ERROR CODE</i>	<i>PURPOSE</i>
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
403	FORBIDDEN	Invalid/missing certificate = provider is not authenticated
501	NOT_IMPLEMENTED	Method not implemented
404	TRANSACTION_MISSING	The requested transaction ID does not exist.

3.2.16.1 MESSAGE ELEMENTS Delete an entered standing order

In case of an unauthorised standing order, no content is returned in the response if the deleting was successful.

In case of an authorised standing order, the response contains a body (specified here below) with the standing order identification and also signId, which is necessary to complete the deleting of the authorised standing order process.

<i>LEVEL</i>	<i>MESSAGE ELEMENT</i>	<i>OCCURRENCE</i>	<i>PAYMENT</i>	<i>FORMAT TYPE</i>	<i>PRESENTATION</i>
--------------	------------------------	-------------------	----------------	--------------------	---------------------

TYPE					
+	standingOrderIdentification	[1..1]	ALL	±	Standing order identification
++	instructionIdentification	[1..1]	ALL	Max35Text	Instruction identification
++	endToEndIdentification	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	End to end identification
++	transactionIdentification	[1..1]	ALL	Max35Text	Transaction identification
+	signInfo	[1..1]	ALL	±	Information on the current instruction authorization
++	state	[1..1]	PISP ALL	Text	Status of the current instruction authorization in a format supported by bank
++	signId	[1..1]	PISP ALL	Text	Unique identifier for the current instruction authorization

3.2.17 Existing standing order change (PUT /my/standingorders/{transactionIdentification})

Resource for change of standing order specification.

Resource characteristics

Mandatory: no

URI: /my/standingorders/{transactionIdentification}

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Standing%20Orders/-updateStandingOrder>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: PUT

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: no

Sorting: no

Filtration: no

Input character set: limitation is based on definitions from chapter 3.2.1

Query parameters of the request: **not defined**

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of PUT request and response for calling, please see Chapters 3.2.17.1 MESSAGE ELEMENTS Standing order change and 3.2.17.2 MESSAGE ELEMENTS Response for standing order change.

Error codes defined for the service PUT Existing standing order change

HTTP STATUS CODE	ERROR CODE	PURPOSE
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
401	UNAUTHORISED	Invalid/missing certificate = provider is not authenticated
403	FORBIDDEN	Calling of the method which does not correspond to the licence, or invalid certificate.

400	FIELD_MISSING	Missing mandatory field in the request.
400	FIELD_INVALID	FIELD value is not valid.
400	AC02	[InvalidDebtorAccountNumber] – invalid account identifier in the request content.
400	AC03	[InvalidCreditorAccountNumber] - recipient account number is closed, blocked, credit card are disabled for the account type, or the recipient's account number is in invalid format (note: only valid for in-house payments).
400	AC10	[InvalidDebtorAccountCurrency] – the specified payer account currency does not correspond to the client's account currency for the account number held in the bank (the account currency is optional, in the case of multicurrency accounts, the account currency must be specified - Raiffeisenbank).
400	AC12	[InvalidAccountType] - account type does not match allowed account types (e.g., a non-paying account).
403	AG01	[TransactionForbidden] – absent consent to access to Balance Check at the account.
400	AM05	[Duplication] - this is a duplicate. Universal code for a duplicate CISP query (validated by, for example, CSOB) or a duplicate payment via GIS (non-unique payment reference).
400	AM11	[InvalidTransactionCurrency] – the request contains a currency not trade/not supported.
400	AM12	[InvalidAmount] – wrong amount. For instance, too low or high amount or wrong number format according to the number of decimal places according to the ISO 4217.
400	FF01	[Invalid File Format] – invalid JSON format or other technical problem with the query processing.
400	BE19	[InvalidChargeBearerCode] - invalid fee type for the given transaction type.
400	DT01	[InvalidDate] - non-existent date value or format. Maturity in the future or in the past. Transactions on a non-business day of a bank (unless the bank receives such orders).
400, 50x	NARR	Narrative – a general reason for rejecting the payment, with an addition of error-related information.
400	RC07	[InvalidCreditorBICIdentifier] – invalid recipient SWIFT / BIC code.
400	RC10	[InvalidCreditorClearingSystemMemberIdentifier] - invalid

		recipient bank code identification.
400	RR03	[MissingCreditorNameOrAddress] - the field is missing the required data about the recipient's name or address (or its part). If the data is in bad format, the error code FIELD_INVALID is used.
400	RR10	[InvalidCharacterSet] – invalid character set in the request.
400	REC_SEND	[RECEIVER_SAME_AS_SENDER] – Payer's account is the same as the payee's account (Applies to all types of payments)
404	TRANSACTION_MISSING	The requested transaction ID does not exist.

3.2.17.1 MESSAGE ELEMENTS Standing order change

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
+	standingOrderIdentification	[1..1]	ALL	Max35Text	Standing order identification
++	instructionIdentification	[1..1]	ALL	Max35Text	Instruction identification
++	endToEndIdentification	[0..0] [1..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	Max35Text	End to end identification
++	transactionIdentification	[0..0]	ALL	Max35Text	Transaction identification
+	paymentTypeInfo	[0..1]	ALL	The same object as to initiate payments chapter ...	
+	amount	[1..1]	ALL	STDO - AmountType3CZ ESST - AmountType3CZ XBST - AmountType3Choice	Payment amount

++	instructedAmount	[1..1]	ALL	CurrencyAndAmount	Amount and currency in the instruction
+++	value	[1..1]	ALL	Amount	Amount of the transfer
+++	currency	[1..1]	ALL	CurrencyCode	Transfer Currency
+	requestedExecutionDate	[1..1]	ALL	ISODate	Requested date of payment
+	standingOrder	[1..1]	ALL	Structure	Structure describing the parameters of the standing orders.
++	alias	[0..1]	ALL	Max250Text	Description resp. the user-defined standing order payment name.
++	execution	[1..1]	ALL		Features of execution a standing order for payment.
+++	mode	[1..1]	ALL	Max35Text	The execution mode defines when or how standing order will be cancelled, processed the last time. Possible values: UNTIL_DATE (standing order is valid until specific date - field lastExecutionDate), UNTIL_CANCELLATION (standing order is valid forever and must be cancelled by client), AFTER_MAX_ITERATION_EXCEEDED (certain count of executions is specified - field maxIterations) or MAX_AMOUNT_EXCEEDED (maximum amount which can

					be transferred for this order is specified, if next iteration would exceed this amount it is not executed - field maxAmount).
+++	modeDue	[1..1]	ALL	Max35Text	The execution due mode defines how the date when order should be executed is specified. Possible values: DUE_DAY_OF_MONTH (specific number of day in the month is defined) or DUE_OR_BEFORE_DAY_OF_MONTH (is defined as a specific day of the month or the previous day if the day falls on a non-banking day) or DUE_OR_NEXT_DAY_OF_MONTH (is defined as a specific day of the month or the next day if the day falls on a non-banking day) or DUE_LAST_DAY_OF_MONTH (order is executed on last day of particular month).
+++	interval	[1..1]	ALL	Max10Text	Execution interval defines how often order is executed. Possible values: DAILY, WEEKLY, MONTHLY, BI_MONTHLY, QUARTERLY, HALFYEARLY, YEARLY, IRREGULAR.

+++	intervalDue	[1..1]	ALL	Number	Value represents order number of the day within particular period when the standing order will be regularly executed. Possible values: 1-7 (for WEEKLY interval), 1-28 for MONTHLY, 1-2 for BI_MONTHLY, 1-3 for QUARTERLY, 1-6 for HALFYEARLY, 1-12 for YEARLY.
++	validity	[0..1]	ALL		Elements restricting the validity of the standing order.
+++	lastExecutionDate	[0..1]	ALL	ISODate	Date when the last order will be processed. Null value responds to ending by the user.
+++	maxIterations	[0..1]	ALL	Number	Maximum number of iterations - processing of the standing order. Only applicable in combination with executionMode AFTER_MAX_ITERATION_EXCEEDED.
+++	maxAmount	[0..1]	ALL	Amount	Maximum amount to be transferred using the standing order. Only applicable in combination with executionMode AFTER_MAX_AMOUNT_EXCEEDED.
++++	value	[1..1]	ALL	Amount	Maximum amount of the transfer
++++	currency	[1..1]	ALL	CurrencyCode	Transfer Currency of the maximum

					amount
++	exceptions	[0..1]	ALL		Elements defining exceptions to executing a standing order.
+++	stoppages	[0..1]	ALL	Array of Max20Text	List of months where there is no payment. Possible values: JANUARY, FEBRUARY, MARCH, APRIL, MAY, JUNE, JULY, AUGUST, SEPTEMBER, OCTOBER, NOVEMBER, DECEMBER
+++	breaks	[0..1]	ALL	Array of Intervals	Break periods.
++++	validFromDate	[1..1]	ALL	ISODate	Start date of one break period. Standing order will not be processed from this date.
++++	validToDate	[1..1]	ALL	ISODate	End date of one break period. Standing order will not be processed to this date.
+	exchangeRateInformation	[0..0]	ALL	The same object as to initiate payments chapter ...	
+	chargeBearer	[0..0] [0..0] [0..1] [0..1]	TUZEM SEPA EHP NONEHP	ChargeBearerType1 Code	Fee payer
+	chargesAccount	[0..0]	ALL	The same object as to initiate payments chapter ...	
+	ultimateDebtor	[0..0] [0..1] [0..0]	TUZEM SEPA EHP	The same object as to initiate payments chapter	

		[0..0]	NONEHP	...	
+	debtor	[0..0]	ALL	The same object as to initiate payments chapter ...	
+	debtorAccount	[1..1]	ALL	The same object as to initiate payments chapter ...	
+	intermediaryAgent1	[0..0]	ALL	The same object as to initiate payments chapter ...	
+	creditorAgent	[0..0] [0..1] [1..1] [1..1]	TUZEM SEPA EHP NONEHP	The same object as to initiate payments chapter ...	
+	creditor	[0..0] [1..1] [1..1] [1..1]	TUZEM SEPA EHP NONEHP	The same object as to initiate payments chapter ...	
+	creditorAccount	[1..1]	ALL	The same object as to initiate payments chapter ...	
+	ultimateCreditor	[0..0] [0..1] [0..0] [0..0]	TUZEM SEPA EHP NONEHP	The same object as to initiate payments chapter ...	
+	purpose	[0..0]	ALL	The same object as to initiate payments chapter ...	
+	instructionForNextAgent	[0..0]	ALL	Instruction code	Instructions for the next bank
+	remittanceInformation	[0..1]	ALL	The same object as to initiate payments chapter ...	

3.2.17.2 MESSAGE ELEMENTS Response for standing order change

The table contains only the elements that appear only in the response message.

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+	standingOrderIdentification	[1..1]	ALL	Max35Text
++	transactionIdentification	[1..1]	Max35Text	Identifier of established standing order instruction
+	paymentTypeInformation	[1..1]	ALL	
++	serviceLevel	[1..1]	±	Service placement (within type of payment)
+++	code	[1..1]	Text	Type of entered payment
+	signInfo	[1..1]	±	Status information and id of unauthorized transactions
++	state	[1..1]	StateCode	Transaction authorization status
++	signId	[1..1]	Text	Identifier of the authorization process of a particular transaction.
+	instructionStatus	[1..1]	Status Code set	Transaction status identifier
+	statusChangeInfo	[0..1]	Text	Specifies the change which was made to the instruction. Used with instructionStatus "ACWC".

Values of serviceLevel.code - The type of entered payment

<i>CODE</i>	<i>DESCRIPTION</i>
DMCT	[DoMestic Credit Transfer] Domestic payment
ESCT	[SEPA Credit Transfer] – SEPA payment
XBCT	[Cross-Border Credit Transfer] – Cross border payment
EXCT	[EEA X-border Credit Transfers] – Cross border payment within the EEP
NXCT	[Non-EEA X-border Credit Transfers] – Cross border payment outside the EEP

Status codes of standing order – StatusCode

<i>HTTP STATUS CODE</i>	<i>STATUS CODE</i>	<i>PURPOSE</i>
200	ACTC	[AcceptedTechnicalValidation] - Authentication and syntactical and semantical validation are successful
200	RJCT	[Rejected] - Payment initiation or individual transaction included in the payment initiation has been rejected
200	ACWC	[AcceptedWithChange] - Instruction is accepted but a change will be made, such as date or remittance not sent

3.2.18 Standing order authorization Step 1 (GET /my/standingorders/{transactionIdentification}/sign/{signId})

First step of standing order authorization.

Resource characteristics

Mandatory: no

URI: /my/standingorders/{transactionIdentification}/sign/{signId}

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Standing%20Orders/-getStandingOrderAuthorizationDetail>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: GET

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: no

Sorting: no

Filtration: no

Input character set: limitation is based on definitions from chapter 3.2.1

Query parameters of the request: **not defined**

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is always mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of GET requests and response for calling, please see Chapter 3.2.16.1 MESSAGE ELEMENTS Step I. Standing order authorization detail

Error codes defined for the service GET step I. standing order authorization detail:

<i>HTTP STATUS CODE</i>	<i>ERROR CODE</i>	<i>PURPOSE</i>
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
403	FORBIDDEN	Invalid/missing certificate = provider is not authenticated
501	NOT_IMPLEMENTED	Method not implemented

404	ID_NOT_FOUND	The required id does not exist
400	AUTH_LIMIT_EXCEEDED	The resource may not be authorized with this method

3.2.18.1 MESSAGE ELEMENTS Step I. Standing order authorization detail

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+	scenarios	[1..1]	±	A set of possible authorization scenarios
+	signInfo	[1..1]	±	Information on instruction authorization
++	state	[1..1]	Text	Status of transaction authorization in a format supported by the bank
++	signId	[1..1]	Text	Unique identifier for current transaction authorization

3.2.19 Standing order authorization Step II (POST /my/standingorders/{transactionIdentification}/sign/{signId})

Second step of standing order authorization.

Resource characteristics

Mandatory: yes

URI: /my/standingorders/{transactionIdentification}/sign/{signId}

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Standing%20Orders/-postStandingOrderAuthorizationInitiation>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: POST

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: no

Sorting: no

Filtration: no

Input character set: limitation is based on definitions from chapter 3.2.1

Query parameters of the request: **not defined**

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is always mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of POST request and response for calling, please see chapter 3.2.17.1 MESSAGE ELEMENTS Step II. Standing order authorization initiation – specific for each bank

Error codes defined for the POST service Step II. Standing order authorization initiation:

<i>HTTP STATUS CODE</i>	<i>ERROR CODE</i>	<i>PURPOSE</i>
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
403	FORBIDDEN	Invalid/missing certificate = provider is not authenticated
404	ID_NOT_FOUND	The required id does not exist
400	AUTH_LIMIT_EXCEEDED	The resource may not be authorized with this method
400	INVALID_AUTHORIZATION_REDIRECT	The value of redirect uri is not valid (not registered)

	_URI	
--	------	--

3.2.19.1 MESSAGE ELEMENTS Step II. Standing order authorization initiation – specific for each bank

Request parameters:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+	authorizationType	[1..1]	Text	Code of required authorization (from authorization scenarios)
+	redirectUrl	[0..1]	Text	URL link or package of federated authorization call back address. This address is used by the federated bank authorization to redirect back to the TPP application after authorization is complete.

Response parameters:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+	authorizationType	[1..1]	±	Code of required <i>authorization (from authorization scenarios)</i>
+	href	[0..1]	±	Reference to call federated authorization
++	url	[1..1]	Text	URL link or package of federated authorization
++	id	[0..1]	Text	Possible id for calling federated authorization
+	method	[0..1]	Text	Method to use href link and federated authorization.
+	formData	[0..1]	±	Optional element. For the method POST federated authorization (authorizationType=USERAGENT_REDIRECT) element Contains data for sending in redirecting to

				federated authorization.
++	SAMLRequest	[0..1]	Text	Optional parameter. For the method POST federated authorization (authorizationType=USERAGENT_REDIRECT) element Contains data of SAML request
++	relayState	[0..1]	Text	Optional parameter. For the method POST federated authorization (authorizationType=USERAGENT_REDIRECT) element Contains relayState for returnable value.
+	signInfo	[1..1]	±	Information on instruction author
++	state	[1..1]	Text	Status of transaction authorization in a format supported by the bank
++	signId	[1..1]	Text	Unique identifier for current transaction authorization

3.2.20 Standing order authorization Step III (PUT /my/standingorders/{transactionIdentification}/sign/{signId})

Third step of standing order authorization.

Resource characteristics

Mandatory: no

URI: /my/standingorders/{transactionIdentification}/sign/{signId}

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Standing%20Orders/-putStandingOrderAuthorizationInitiation>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: PUT

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: no

Sorting: no

Filtration: no

Input character set: limitation is based on definitions from chapter 3.2.1

Query parameters of the request: **not defined**

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is always mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of PUT request and response for calling, please see Chapter 3 3.2.20.1 MESSAGE ELEMENTS Step III. Standing order authorization finalization - specific for each bank

Error codes defined for the PUT service Step III. Payment authorization initiation:

<i>HTTP STATUS CODE</i>	<i>ERROR CODE</i>	<i>PURPOSE</i>
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
403	FORBIDDEN	Invalid/missing certificate = provider is not authenticated
501	NOT_IMPLEMENTED	Method not implemented
404	ID_NOT_FOUND	The required id does not exist
400	AUTH_LIMIT_EXCEEDED	The resource may not be authorized with this method

3.2.20.1 MESSAGE ELEMENTS Step III. Standing order authorization finalization - specific for each bank

Request parameters

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+	authorizationType	[1..1]	Text	Code of required authorization (from authorization scenarios)
+	oneTimePassword	[0..1]	Max35Text	One-time password received e.g. in SMS.

Response parameters

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+	State	[1..1]	Text	Status of transaction authorization in a format supported by the bank
+	instructionStatus	[0..1]	StatusCode	Status of entered standing order
+	pollInterval	[0..1]	Number	Value in milliseconds how often I can ask for authorization status.

Status codes of standing order – StatusCode

HTTP STATUS CODE	STATUS CODE	PURPOSE
200	ACTC	[AcceptedTechnicalValidation] - Authentication and syntactical and semantical validation are successful
200	RJCT	[Rejected] - Payment initiation or individual transaction included in the payment initiation has been rejected
200	ACWC	[AcceptedWithChange] - Instruction is accepted but a change will be made, such as date or remittance not sent

3.2.21 Batch payment - bulk authorization

Overview of resources for the establishment, authorization and administration of batch payments.

3.2.22 New batch payment - instruction initiation (POST /my/batchpayments)

Resource for entering a new batch payment.

Resource characteristics

Mandatory: yes

URI: /my/batchpayments

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Batch%20Payments/-postNewBatchPayment>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: POST

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: no

Sorting: no

Filtration: no

Input character set: limitation is based on definitions from chapter 3.2.1

Query parameters of the request: **not defined**

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is always mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of PUT request and response for calling, please see Chapter 3.2.22.1 MESSAGE ELEMENTS New batch payment initiation and 3.2.22.2 MESSAGE ELEMENTS Response for new batch payment initiation.

Error codes defined for the service POST Query for Batch payment initiation

<i>HTTP STATUS CODE</i>	<i>ERROR CODE</i>	<i>PURPOSE</i>
-----------------------------	-------------------	----------------

401	UNAUTHORISED	Missing certificate.
403	FORBIDDEN	Calling of the method which does not correspond to the licence, or invalid certificate.
400	FIELD_MISSING	Missing mandatory field in the request.
400	FIELD_INVALID	FIELD value is not valid.
400	FF01	[Invalid File Format] – invalid JSON format or other technical problem with the query processing.
400, 50x	NARR	Narrative – a general reason for rejecting the payment, with an addition of error-related information.
400	RR10	[InvalidCharacterSet] – invalid character set in the request.
400	ACC_DIFFERENT	Requested payments are not initiated from the same source account.
400	CURR_DIFFERENT	Requested payments are not in the same currency.
400	TYPE_DIFFERENT	Requested payments do not have the same serviceLevel.
400	TOO_MANY_PMTS	Maximum amount of payments in one batch exceeded.
400	INCORRECT_SIGNID	One or more signIDs are missing or are incorrect or do not have open state.

3.2.22.1 MESSAGE ELEMENTS New batch payment initiation

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+	exchangeIdentification	[1..1]	Max18Text	Clear query identification
+	instructionName	[0..1]	Max35Text	An entry field for an optional batch name
+	payments	[1..1]	±	Collection of payments
++	signId	[1..n]	Max35Text	Identifier of the authorization process of a particular transaction.

3.2.22.2 MESSAGE ELEMENTS Response for new batch payment initiation

The table contains only the elements that appear only in the response message.

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+	transactionIdentification	[1..1]	Max35Text	Identifier of established batch payment instruction
+	signInfo	[1..1]	±	Status information and id of unauthorized transactions
++	State	[1..1]	StateCode	Transaction authorization status
++	signId	[1..1]	Text	Identifier of the authorization process of a batch transaction.
+	instructionStatus	[1..1]	Status Code set	Transaction status identifier
+	statusChangeInfo	[0..1]	Text	Specifies the change which was made to the instruction. Used with instructionStatus "ACWC".

Status codes of batch payment – StatusCode

HTTP STATUS CODE	STATUS CODE	PURPOSE
200	ACTC	[AcceptedTechnicalValidation] - Authentication and syntactical and semantical validation are successful
200	RJCT	[Rejected] - Payment initiation or individual transaction included in the payment initiation has been rejected

200	ACWC	[AcceptedWithChange] - Instruction is accepted but a change will be made, such as date or remittance not sent
-----	------	---

3.2.23 Batch payment delete (DELETE) /my/batchpayments/{transactionIdentification}{?allInstructions}

A resource for deleting the unauthorised batch payment. Deleting is not conditioned by the transaction authorisation as it is not a payment received by the bank.

Resource characteristics

Mandatory: no

URI: /my/batchpayments/{transactionIdentification}{?allInstructions}

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Batch%20Payments/-deleteBatchPayment>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: DELETE

Authorization: request **requires** authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: no

Sorting: no

Filtration: no

Input character set: no restrictions in relation to chapter 3.2.1

Query parameters of the request:

PARAMETER	TYPE	MANDATORY	PURPOSE
allInstructions	Boolean	No	If this parameter is set to true, not only will the bulk authorization be deleted, but also all individual instructions which were a part of the bulk.

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

Error codes defined for the service DELETE the entered unauthorised batch payment:

HTTP STATUS CODE	ERROR CODE	PURPOSE
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
403	FORBIDDEN	Invalid/missing certificate = provider is not authenticated
501	NOT_IMPLEMENTED	Method not implemented
404	TRANSACTION_MISSING	The requested transaction ID does not exist.

3.2.24 Batch payment status (GET /my/batchpayments/{transactionIdentification}/status)

Resource for getting status of the batch payment

Resource characteristics

Mandatory: yes

URI: /my/batchpayments/{transactionIdentification}/status

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Batch%20Payments/-getBatchPaymentStatus>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: GET

Authorization: request **requires no** authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: no

Sorting: no

Filtration: no

Input character set: no restrictions in relation to chapter 3.2.1

Query parameters of the request: **not defined**

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is **not** mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

Error codes defined for the service GET status of entered/initiated batch pyment:

<i>HTTP STATUS CODE</i>	<i>ERROR CODE</i>	<i>PURPOSE</i>
401	UNAUTHORISED	Invalid/missing certificate = provider is not authenticated
404	TRANSACTION_MISSING	The requested transaction ID does not exist.

3.2.24.1 MESSAGE ELEMENTS Status of initiated batch payment

<i>LEVEL</i>	<i>MESSAGE ELEMENT</i>	<i>OCCURRENCE</i>	<i>FORMAT TYPE</i>	<i>PRESENTATION</i>
+	instructionStatus	[1..1]	StatusCode	Status of entered batch payment
+	payments	[0..n]	±	Collection of payments
++	transactionIdentification	[1..1]	Max35Text	Identifier of established individual transaction
++	signInfo	[1..1]	±	Status information and id

				of individual transaction
+++	State	[1..1]	StateCode	Individual transaction authorization status
+++	signId	[0..1]	Text	Identifier of the authorization process of a batch transaction.

Status codes of batch payment – StatusCode

HTTP STATUS CODE	STATUSCODE	PURPOSE
200	ACTC	[AcceptedTechnicalValidation] - Authentication and syntactical and semantical validation are successful
200	RJCT	[Rejected] - Payment initiation or individual transaction included in the payment initiation has been rejected
200	ACSP	[AcceptedSettlementInProgress] - All preceding checks such as technical validation and customer profile were successful and therefore the payment initiation has been accepted for execution

3.2.25 Batch payment authorization Step I (GET /my/batchpayments/{transactionIdentification}/sign/{signId})

First step of standing order authorization.

Resource characteristics

Mandatory: no

URI: /my/batchpayments/{transactionIdentification}/sign/{signId}

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Batch%20Payments/-getBatchPaymentAuthorizationDetail>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: GET

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: no

Sorting: no

Filtration: no

Input character set: no restrictions in relation to chapter 3.2.1

Query parameters of the request: **not defined**

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of GET response for calling, please see Chapter 3.2.25.1 MESSAGE ELEMENTS Step I. Batch payment authorization detail

Error codes defined for the service GET step I. batch payment authorization detail:

<i>HTTP STATUS CODE</i>	<i>ERROR CODE</i>	<i>PURPOSE</i>
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
403	FORBIDDEN	Invalid/missing certificate = provider is not authenticated
501	NOT_IMPLEMENTED	Method not implemented
404	ID_NOT_FOUND	The required id does not exist
400	AUTH_LIMIT_EXCEEDED	The resource may not be authorized with this method

3.2.25.1 MESSAGE ELEMENTS Step I. Batch payment authorization detail

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+	Scenarios	[1..1]	±	A set of possible authorization scenarios
+	signInfo	[1..1]	±	Information on instruction authorization
++	State	[1..1]	Text	Status of transaction authorization in a format supported by the bank
++	signId	[1..1]	Text	Unique identifier for current transaction authorization

3.2.26 Batch payment authorization Step II (POST /my/batchpayments/{transactionIdentification}/sign/{signId})

Second step of batch payment authorization.

Resource characteristics

Mandatory: yes

URI: /my/batchpayments/{transactionIdentification}/sign/{signId}

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Batch%20Payments/-postBatchPaymentAuthorizationInitiation>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: POST

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: no

Sorting: no

Filtration: no

Input character set: no restrictions in relation to chapter 3.2.1

Query parameters of the request: **not defined**

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of POST request and response for calling, please see chapter 3.2.26.1 MESSAGE ELEMENTS Step II. Batch payment authorization initiation - specific for each bank

Error codes defined for the POST service Step II. Batch payment initiation:

HTTP STATUS CODE	ERROR CODE	PURPOSE
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
403	FORBIDDEN	Invalid/missing certificate = provider is not authenticated
404	ID_NOT_FOUND	The required id does not exist
400	AUTH_LIMIT_EXCEEDED	The resource may not be authorized with this method
400	INVALID_AUTHORIZATION_REDIRECT_URI	The value of redirect uri is not valid (not registered)

3.2.26.1 MESSAGE ELEMENTS Step II. Batch payment authorization initiation – specific for each bank

Request parameters:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+	authorizationType	[1..1]	Text	Code of required authorization

				(from authorization scenarios)
+	redirectUrl	[0..1]	Text	<p>URL link or package of federated authorization call back address.</p> <p>This address is used by the federated bank authorization to redirect back to the TPP application after authorization is complete.</p>

Response parameters:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+	authorizationType	[1..1]	±	Code of required <i>authorization (from authorization scenarios)</i>
+	Href	[0..1]	±	Reference to call federated authorization
++	url	[1..1]	Text	URL link or package of federated authorization
++	Id	[0..1]	Text	Possible id for calling federated authorization
+	Method	[0..1]	Text	Method to use href link and federated authorization.
+	formData	[0..1]	±	<p>Optional element. For the method POST federated authorization (authorizationType=USERAGENT_REDIRECT) element</p> <p>Contains data for sending in redirecting to federated authorization.</p>
++	SAMLRequest	[0..1]	Text	Optional parameter. For the method POST federated authorization (authorizationType=USERAGENT_REDIRECT) element

				Contains data of SAML request
++	relayState	[0..1]	Text	Optional parameter. For the method POST federated authorization (authorizationType=USERAGENT_REDIRECT) element Contains relayState for returnable value.
+	signInfo	[1..1]	±	Information on instruction author
++	State	[1..1]	Text	Status of transaction authorization in a format supported by the bank
++	signId	[1..1]	Text	Unique identifier for current transaction authorization

3.2.27 Batch payments authorization Step III (PUT /my/batchpayments/{transactionIdentification}/sign/{signId})

Third step of batch payment authorization.

Resource characteristics

Mandatory: no

URI: /my/batchpayments/{transactionIdentification}/sign/{signId}

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Batch%20Payments/putBatchPaymentAuthorizationInitiation>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: PUT

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: no

Sorting: no

Filtration: no

Input character set: no restrictions in relation to chapter 3.2.1

Query parameters of the request: **not defined**

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of PUT request and response for calling, please see Chapter 3.2.27.1 MESSAGE ELEMENTS Step III. Batch payment authorization finalization - specific for each bank

Error codes defined for the PUT service Step III. Batch payment authorization initiation:

<i>HTTP STATUS CODE</i>	<i>ERROR CODE</i>	<i>PURPOSE</i>
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
403	FORBIDDEN	Invalid/missing certificate = provider is not authenticated
501	NOT_IMPLEMENTED	Method not implemented
404	ID_NOT_FOUND	The required id does not exist
400	AUTH_LIMIT_EXCEEDED	The resource may not be authorized with this method

3.2.27.1 MESSAGE ELEMENTS Step III. Batch payment authorization finalization - specific for each bank

Request parameters

<i>LEVEL</i>	<i>MESSAGE ELEMENT</i>	<i>OCCURRENCE</i>	<i>FORMAT TYPE</i>	<i>PRESENTATION</i>
+	authorizationType	[1..1]	Text	Code of required authorization (from authorization scenarios)

+	oneTimePassword	[0..1]	Max35Text	One-time password received e.g. in SMS.
---	-----------------	--------	-----------	---

Response parameters

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+	State	[1..1]	Text	Status of transaction authorization in a format supported by the bank
+	instructionStatus	[0..1]	StatusCode	Status of entered batch payment
+	pollInterval	[0..1]	Number	Value in milliseconds how often I can ask for authorization status.

Status codes of batch payment – StatusCode

HTTP STATUS CODE	STATUS CODE	PURPOSE
200	ACTC	[AcceptedTechnicalValidation] - Authentication and syntactical and semantical validation are successful
200	RJCT	[Rejected] - Payment initiation or individual transaction included in the payment initiation has been rejected
200	ACWC	[AcceptedWithChange] - Instruction is accepted but a change will be made, such as date or remittance not sent

3.2.28 Get list of authorizations (GET /my/authorizations)

Resource to obtain a list of authorization requests for authenticated user and TPP.

Resource characteristics

URI: `/my/authorizations`

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Payment%20Initiation/-getListOfAuthorizations>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: GET

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: no

Sorting: no

Filtration: no

Input character set: no restrictions in relation to chapter 3.2.1

Query parameters of the request: **not defined**

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of GET request and response for calling, please see Chapter 3.2.28.1 MESSAGE ELEMENTS New payment - payment initiation

Error codes defined for the service POST Query for Payment Initiation

<i>HTTP STATUS CODE</i>	<i>ERROR CODE</i>	<i>PURPOSE</i>
401	UNAUTHORISED	Missing certificate.
403	FORBIDDEN	Calling of the method which does not correspond to the licence, or invalid certificate.

3.2.28.1 MESSAGE ELEMENTS Request to list of authorizations

This service has no content request.

3.2.28.2 MESSAGE ELEMENTS Response to list of authorizations

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+	authorizations	[1..1]	±	Collection of pending authorizations of authenticated user
++	Type	[1..1]	InstructionType	Type of authorized Instruction
++	transactionIdentification	[1..1]	Max35Text	Identifier of established transaction
++	validUntil	[0..1]	ISODateTime	Date and time when the authorization event for this instruction expires.
++	scenarios	[1..1]	±	A set of possible authorization scenarios
++	signInfo	[1..1]	±	Information on instruction authorization
+++	State	[1..1]	Text	Status of transaction authorization in a format supported by the bank
+++	signId	[1..1]	Text	Unique identifier for current transaction authorization

INSTRUCTION TYPE CODE	DESCRIPTION
PAYMENT	Single payment authorization
STANDINGORDER	Recurring payment authorization

3.2.29 List of client's awaiting payments (GET /my/payments/awaiting{?size,page,sort,order})

Paged list of client's awaiting payments. Each payment contains a unique id usable for URI referencing, e.g. for the payment detail.

Additionally, the debtorAccount contains an API account id to call the account details resource to which the payment order is issued.

Resource characteristics

Mandatory: no

URI: /my/payments/awaiting/{?size,page,sort,order}

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Account%20information/-getAccountAwaitingTransactions>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: GET

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: yes

Sorting: yes

Filtering: no

Input character set: no restrictions in relation to chapter 3.1

Query parameters of the request:

PARAMETER	TYPE	MANDATORY	PURPOSE
size	Number	No	Paging. Number of entries per page
page	Number	No	Paging. Required page. + Default: 0
sort	Text	No	A list of fields separated by comma for sorting, arranged according to the meaning
order	Text	No	A list of arrangement methods (ASC, DESC) separated by comma. The order corresponds to the order of fields in the sort parameter.sort parameter.

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is always mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers.](#)

The content of GET response for calling, please see Chapter 3.2.29.1 RESPONSE MESSAGE ELEMENTS List of client’s awaiting payments

Error codes defined for the GET service, List of client’s awaiting payments:

<i>HTTP STATUS CODE</i>	<i>ERROR CODE</i>	<i>PURPOSE</i>
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
401	UNAUTHORISED	Invalid/missing certificate = provider is not authenticated
403	FORBIDDEN	Authentication with an invalid certificate or expired access token, a call that does not match a third-party license.
400	PAGE_NOT_FOUND	Request for a non-existing page
400	PARAMETER_INVALID	The value of the parameter is not valid

3.2.29.1 MESSAGE ELEMENTS Response for GET list of client’s awaiting payments

<i>LEVEL</i>	<i>MESSAGE ELEMENT</i>	<i>OCCURRENCE</i>	<i>FORMAT TYPE</i>	<i>PRESENTATION</i>
+	payments	[1..1]	±	Payment collection
++	paymentIdentification	[1..1]	Max35Text	Payment identification
+++	instructionIdentification	[1..1]	Max35Text	Instruction identification
+++	transactionIdentification	[1..1]	Max35Text	Transaction identification
+++	endToEndIdentification	[0..1]	Max35Text	End to end identification

++	paymentTypeInformation	[1..1]	PaymentTypeInformation19	Payment type information
+++	instructionPriority	[0..1]	Priority2Code	Instruction priority
+++	serviceLevel	[1..1]	ServiceLevel8CZ	Service level
++++	Code	[0..0]	ExternalServiceLevel1Code	Service level code
+++	categoryPurpose	[0..0]	CategoryPurpose1Choice	Category of the purpose of payment
++++	Code	[0..0]	ExternalCategoryPurpose1Code	Payment purpose category code
++++	proprietary	[0..0]	Max35Text	Free format for the purpose of the payment purpose
++	amount	[1..1]	The same object as to initiate payments chapter ...	Payment amount
++	requestedExecutionDate	[0..1]	ISODate	Requested date of payment
++	debtorAccount	[1..1]	±	Payer account
+++	id	[1..1]	Max35Text	API Payment account identifier
+++	identification	[1..1]	AccountIdentification4ChoiceCZ	Identification of the payer's account
++++	Iban	[1..1]	IBAN2007Identifier	Payer account number in IBAN format. either this parameter or other
++++	Other	[1..1]	GenericAccountIdentification1CZ	Payer account number in other/local format. either this parameter or iban
+++++	identification	[1..1]	Max35Text	Payer account number in local BBAN format
+++	currency	[0..1]	CurrencyCode ISO 4217	Payer's account currency
++	creditorAgent	[0..0]	The same object as to	Bank of receiver

			initiate payments chapter ...	
++	creditorAccount	[1..1]	The same object as to initiate payments chapter ...	Payee account
++	remittanceInformation	[0..1]	The same object as to initiate payments chapter ...	Payment information
++	awaitingStatus	[1..1]	FUTURE - payment waiting for execution date. OVERDUE - for example, in case of insufficient funds at the account OTHER - other kind of payment status	Awaiting status of payment.

3.2.30 Payment initialization with one SCA

3.2.30.1 Introduction

This article aims to extend the Czech Standard for Open Banking 6.0 [1] to cover the payment with one SCA option, since two SCAs are an obstacle from EBA's perspective. The focus is to follow standards with backward compatibility with the previous versions of COBS API.

EBA announced in 2020 that two SCAs for payment initialization is generally considered to be an obstacle [2] when all payment data is sent from TPP and therefore it can be marked as an issue during the bank audit. The same view comes from "European Third Party Providers Association" [3] and National Bank of Slovakia [4]. Czech National Bank gave penalty to UCB for two reasons in year 2022, one of them was 2-SCA issue [5]

3.2.30.2 Normative references

RAR 1.0 [6] [7]

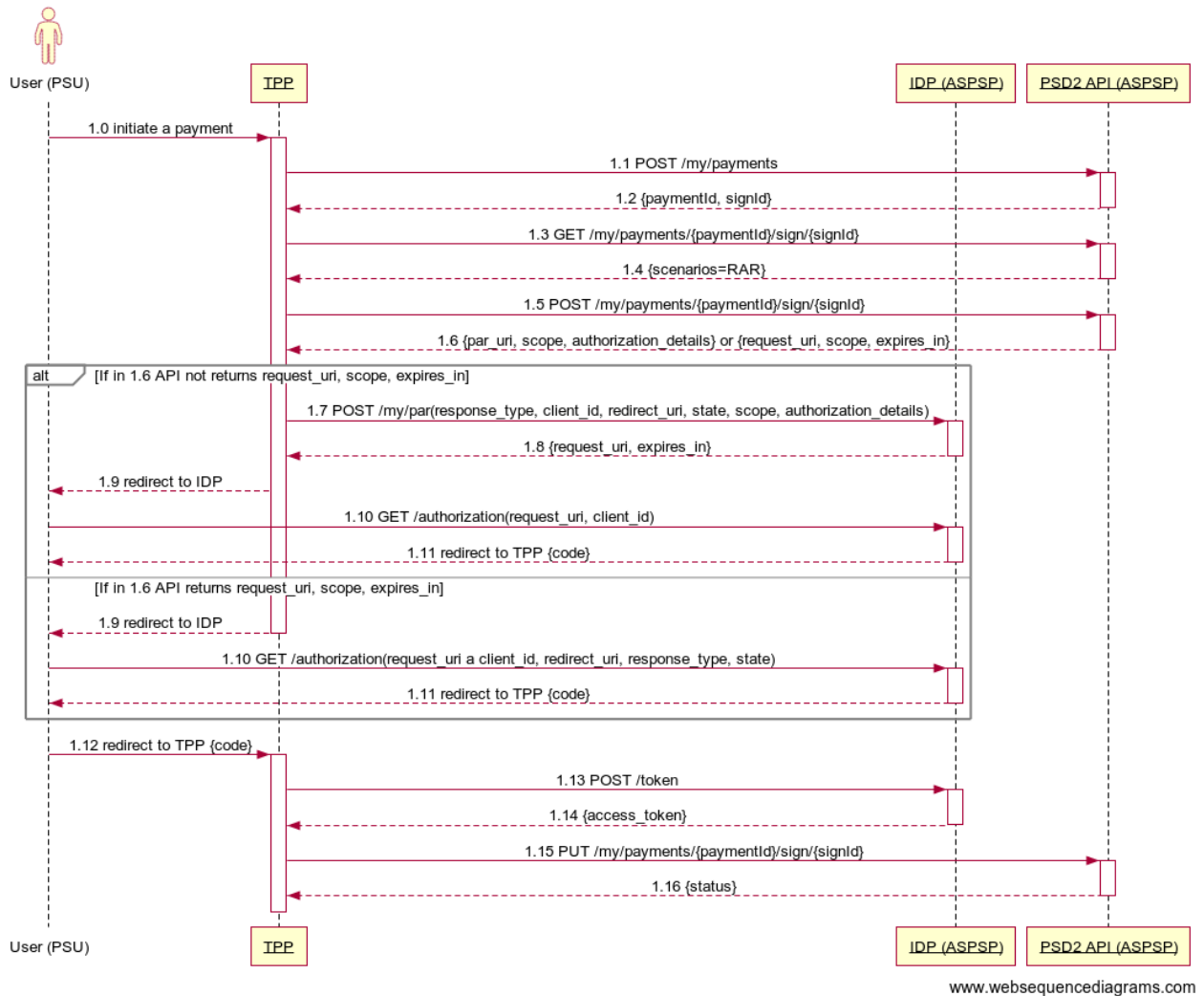
PAR [8] (optional)

JARM [9] (optional)

JAR [10] (optional)

3.2.30.3 Main flow

OneSCA main flow



Payment with one SCA is based on Rich Authorization Request [7] [6]

1.1 TPP sends payment data to PSD2 API.

PSD2 API makes all validations that can be done without knowing the PSU identity-

1.2 PSD2 API retrieves back paymentId and signId

It is expected but not required that encrypted data is JWE. Encrypted object must contain signId

1.3 TPP asks for available sign scenarios

1.4 New sign scenario RAR is returned

1.5 TPP asks for sign procedure details

1.6 The PSD2 API returns a PAR endpoint and optionally custom scope or optionally encrypted object "authorization_details" that contains the information of payment data needed for SCA.

Custom scope containing signId creates a relationship between the payment and access token.

If an encrypted object is not returned, payment information is expected to be sent to the IDP via a callback.

If the scope or encrypted object are present in response, the TPP must use them for par or authorization request.

- a. If response 1.6 **does not contain** request_uri, scope, expires_in parameters, the sequence continues with step 1.7
- b. If response 1.6 **contains the parameters** request_uri, scope, expires_in, the sequence continues with step 1.9, and request 1.10 contains the parameters request_uri, client_id, redirect_uri, response_type and state (from response 1.6)

1.7 TPP pushes OAuth2 authorization data to IDP by using previously returned PAR endpoint.

Authorization data also contains custom scope and encrypted object.

IDP is capable to decrypt the encrypted object.

1.8 In response TPP returns request_uri and expires_in

Request_uri is meaningless reference to encrypted object. It is not required to be directly addressed. Expires_in is validity period of request_uri in seconds.

1.9 TPP redirects PSU's device to IDP

1.10 IDP displays payment information to PSU and provides SCA and payment authorization at once

1.11 The success SCA leads to code that is redirects back to TPP

1.13 TPP exchanges code with access token

The token is bind to signId.

1.15 TPP uses access token to finalize the operation.

PSD2 API must validate if the access token is tied to the signId.

PSD2 API must implement protection against repeated attempts to complete the payment with the same signId.

3.2.30.4 List of PSD2 API required validations

API technical specifications: <https://github.com/Czech-BA/COBS/tree/feature/payment-with-one-SCA>

- Access to PSD2 API is protected by MTLS with qualified client certificate (QWAC, ETSI 1.3.1+).
- PSD2 API must validate if the access token is bind to the signId.
- PSD2 API must implement protection against repeated attempts to complete the payment with the same access token.

3.2.30.5 Additional information

- The way how to bind the access token with signId is up to particular solutions. It can be done by using custom scopes or by extending IDP introspect endpoint with encrypted object. The particular solution is up to ASPs.
- PAR is highly recommended, but it is not required. The par request can be avoided by using directly /authorization endpoint
- JAR is recommended but it is not required. PAR endpoint can be called with traditional query parameters like response_type, state, client_id, redirect_uri, code_challenge, code_challenge_method, scope
- The preferred authentication methods of IDP /token and /par endpoints are private_key_jwt, tls_client_auth, self_signed_tls_client_auth, but for backward compatibility COBS allows to use client_secret_basic, client_secret_post, client_secret_jwt

- JARM is recommended but it is not required. If the query parameter `response_mode` is not present than code is returned in query parameter of redirect URL

3.2.30.6 Example of message calls

For a detailed specification, you can look at the current description in the CBA repository for the Czech Open Banking Standard on GitHub here <https://github.com/Czech-BA/COBS/tree/feature/payment-with-one-SCA>

New payment - payment initiation (POST /my/payments):

**Header Authorization is optional and it is up to each of the banks to allow combined flow authorizations for payment with an already obtained token.*

```

POST /my/payments HTTP/1.1
Accept: application/json
X-Request-ID: fead8a62-e5ae-4184-be19-905b281a1061
User-Involved: true
TPP-Name: api2bank
Content-Type: application/json
Authorization: Bearer 4a58d892-4db5-477d-b92f-fa897d8093fb
{
  "paymentIdentification": {
    "instructionIdentification": "LC platba"
  },
  "amount": {
    "instructedAmount": {
      "value": 1260,
      "currency": "CZK"
    }
  },
  "debtorAccount": {
    "identification": {
      "iban": "CZ6858000000000000999999"
    }
  },
  "creditorAccount": {
    "identification": {
      "iban": "CZ6858000000000000999999"
    }
  },
  "remittanceInformation": {
    "unstructured": "VS/7418529630/SS/1234567890"
  }
}

```

New payment – payment initiation response:

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
X-Request-ID: fead8a62-e5ae-4184-be19-905b281a1061

{
  "paymentIdentification": {
    "instructionIdentification": "LC platba",
    "transactionIdentification": "db8f4657-d874-4926-b8ff-3f521fea670c"
  },
  "serviceLevel": {
    "code": "DOMESTIC"
  },
  "signInfo": {
    "state": "OPEN",
    "signId": "7c44e312-5019-429a-841f-d9800be8688d"
  },
  "instructionStatus": "ACTC"
}
```

Step I. Payment authorization detail (GET /my/payments/{paymentId}/sign/{signId}) request:

**Header Authorization is optional and it is up to each of the banks to allow combined flow authorizations for payment with an already obtained token.*

```
GET /my/payments/db8f4657-d874-4926-b8ff-3f521fea670c/sign/7c44e312-5019-429a-841f-d9800be8688d HTTP/1.1
Accept: application/json
X-Request-ID: 73cf4f90-25dc-43c9-9bb2-49f1d3d2c33a
User-Involved: true
TPP-Name: api2bank
Authorization: Bearer 4a58d892-4db5-477d-b92f-fa897d8093fb
```

Step I. Payment authorization detail (GET /my/payments/{paymentId}/sign/{signId}) response:

```
HTTP/1.1 200
Content-Type: application/json
X-Request-ID: 73cf4f90-25dc-43c9-9bb2-49f1d3d2c33a
{
  "scenarios": [
    "RAR"
  ],
  "signInfo": {
    "state": "OPEN",
    "signId": "7c44e312-5019-429a-841f-d9800be8688d"
  }
}
```

Step II. Payment authorization initiation - specific for each bank (POST /my/payments/{paymentId}/sign/{signId}) request:

**Header Authorization is optional and it is up to each of the banks to allow combined flow authorizations for payment with an already obtained token.*

```
POST /my/payments/db8f4657-d874-4926-b8ff-3f521fea670c/sign/7c44e312-5019-429a-841f-d9800be8688d HTTP/1.1
Content-Type: application/json
X-Request-ID: c1d9fe4c-12e3-4cad-8bb9-e0abfda41c39
User-Involved: true
TPP-Name: api2bank
Authorization: Bearer 4a58d892-4db5-477d-b92f-fa897d8093fb
{
  "authorizationType": "RAR"
}
```

Step II. Payment authorization initiation - specific for each bank (POST /my/payments/{paymentId}/sign/{signId}) response:

```
HTTP/1.1 200
Content-Type: application/json
X-Request-ID: c1d9fe4c-12e3-4cad-8bb9-e0abfda41c39
Authorization: Bearer 4a58d892-4db5-477d-b92f-fa897d8093fb
{
  "authorizationType": "RAR",
  "rar": {
    "par_uri": "https://as.jtbank.cz/par",
    "scope": "signId7c44e312-5019-429a-841f-d9800be8688d",
    "authorization_details":
      "U2FsdGVkX1/xaq5hQD/PU6JHEncGMTkqtrWwyyzD0JESLsVIsN1m1f0zYYeJBHhg3ghW
      I3BuSm9eZeCYrMbj2VkmPW7rEFX0mI92bb7EZdQe9rmYfCO5nI+1Q7jow38seZAGe2xoa
      Xbrf47YaDZTNdUfFNkhQoEQaa1CjuhXuj3/bvpcj3dtTzj4KExI/a1od6LmiiPP5xch/P
      0E3xGraclZW7QMNMQfSzCuZT7vWyHNAX9yKW8pPb+M0NudCCSvpcCE4hsjijhtIu0PqI0
      oNw75wEkYdt2UrWN2pB2aOrB/hZiuYnkjrABmeerAfit7pK4mLE9xr9KVqSpnMI/TFcmY
      /SUEAE1UBr1SI371GFbSuIPCAdk7KEYJeqJL9AibilMNH5jlniRrNt8gaHf6DmaInOsiB
      LwRM/s4Xprqgnm5PzoEEeWYu8Km05Z/kDtLLA9yHRFiA0oDSScD7bnsnRtSx4cTX1I1T3
      qIop8HfQFd/41hWZfStQFCpC53Ry/fPGfRazkJ5fg8M+RIRd/bOSfDJovF6cg/y1SQUW1
      FRQzBXwuPTxYT6kaSaoAaZeuhnMeQyhX25wmDBDLrUsVFjM0z3eTQWwNqa3jHpmWRlg2K
      lU9Rq6G+cBFQcUuq4Y6/r+aAyhjDE2JmK74SLCDqm3W7jTjMxkrq8KekIR8vhPlqxZcFF
      SmU/cIo1ICkkKUGsT+pMRtBjkIoWXT2ievQ0RHTQhQ8Z1NjHYVkrNMdcEB3Ckuf368901
      gjC+RXEd2/85SBGYLAMiemq/wuWlRMvmzeFoeY9GkLm7ZKwUuTzKm3Y1+c7Jq/9NJ0Hen
      81FEec8GUR+23dXZdbwbPfnSfgtnQvYYBaMm0w4vrreKXdykKYGG300L55mp4NaaNkJ4L
      5nmrUUVf5fM5Y49KV66fqV4rVnnaQORoseuyWQz5HoLVM4ePxKz4xEPZztbanu+R8x6/
      lg5q839o6o56K9Um0nWVcP88UkqAULeQjabr8R89pcAsk7+SpGRqGLlgqeX+NasjsAO7o
      F5YrKBRqme1uw1FOP5KeHOJpAwoq5CTuRyJ7Gd2MjON4SWrk7vpEueJZYAVyIHabQZ6mB
```

```
XcnspmIngqi fD746JFC+kCu4QaeHEKM1MJyUvMRC1zNJ/QLbcq6mFjk53HIMH6uibP3Ty
Yg=="
    }
}
```

Pushed authorization request:

```
POST /par HTTP/1.1
Host: as.jtbank.cz
Content-Type: application/x-www-form-urlencoded
Accept: application/json
X-Request-ID: 9bd13eab-eca2-4d33-bee0-43049862e05b
Authorization: Basic dGVzdGNsaWVudGlkOnRlc3RjbGl1bnRzZWNYZXQ=

    response_type=code&state=af0ifjsldkj&client_id=4e6c2ae9-9954-
42d9-b81d-46305b986cec
    &redirect_uri=https%3A%2F%2Ftpp.fintech.cz%2Fcb
    &code_challenge=K2-ltc83acc4h0c9w6ESC_rEMTJ3bww-uCHaoeK1t8U
    &code_challenge_method=S256&scope=PISP%20signId7c44e312-5019-
429a-841f-d9800be8688d

&authorization_details=U2FsdGVkX1/xaq5hQD/PU6JHEncGMTkqtrWwyyzD0JESLs
VIIsNlmlf0zYYeJBHhg3ghWI3BuSm9eZeCYrMbj2VkmPW7rEFX0mI92bb7EZdQe9rmYfCO
5nI+1Q7jow38seZAGe2xoaXbrf47YaDZTNdUfFNkHQoEQaa1CjuhXuj3/bvpcj3dtTzj4
KExI/alod6LmiiPP5xch/P0E3xGraclZW7QNMmqFSzCuZT7vWyHNAX9yKW8pPb+M0NudC
CSvpdCE4hsjijhtIu0PqI0oNw75wEkYdt2UrWN2pB2aOrB/hZiuYnkjrABmeerAfit7pK
4mLE9xr9KVqSpnMI/TFcmY/SUeAE1UBr1SI371GFbSuIPCAdk7KEYJeqJL9AibilMNH5j
lniRrNt8gaHf6DmaInOsiBLwRM/s4Xprqgnm5PzoEEeWYu8Km05Z/kDtLLA9yHRFiA0oD
SScD7bnsnRtSx4cTX1I1T3qIop8HfQFd/41hWZfStQFCpC53Ry/fPGfRazkJ5fg8M+RIR
d/bOSfDjovF6cg/y1SQUW1FRQzBXwuPTxYT6kaSaoAaZeuhnMeQyhX25wmDBDLrUsVFjm
0z3eTQWwnQa3jHpmWRlg2K1U9Rq6G+cBFQcUuq4Y6/r+aAyhjDE2JmK74SLCDqm3W7jTj
Mxkrq8KekIR8vhPlqxZcFFSmU/cIoLiCkkKUGsT+pMRtBjkIoWXT2ievQ0RHTQhQ8Z1Nj
HYVkrNMdcEB3Ckuf36890lgjC+RXEd2/85SBGYLAMiemq/wuWlRMVmzeFoeY9GkLm7ZKw
UuTzKm3Y1+c7Jq/9NJ0Hen81FEec8GUR+23dXZdbwbPfnSfgtnQvYYBaMm0w4vrreKXdy
kKYGG300L55mp4NaaNkJ4L5nmrUUVf5fM5Y49KV66fqV4rVnnaQORoseuyWQz5HoLVM4
ePxKz4xEPZztbanu+R8x6/lg5q839o6o56K9Um0nWVcP88UkqAULeQjabr8R89pcAsk7+
SpGRqGLlgqeX+NasjsAO7oF5YrKBRqmeluwlFOP5KeHOJpAwoq5CTuRyJ7Gd2MjON4SWR
k7vpEueJZYAVyIHabQZ6mBXcnspmIngqi fD746JFC+kCu4QaeHEKM1MJyUvMRC1zNJ/QL
bcq6mFjk53HIMH6uibP3TyYg==
```

Pushed authorization response:

```
HTTP/1.1 201 Created
Content-Type: application/json
X-Request-ID: 9bd13eab-eca2-4d33-bee0-43049862e05b
Cache-Control: no-cache, no-store
```



```
{
  "request_uri": "urn:uuid:5cc540a9-feca-4ae3-bc55-7aa71fb33a39",
  "expires_in": 60
}
```

Authorization request:

```
GET /authorization?client_id=4e6c2ae9-9954-42d9-b81d-
46305b986cec&request_uri=urn:uuid:5cc540a9-feca-4ae3-bc55-
7aa71fb33a39 HTTP/1.1
Host: as.jtbank.cz
```

Authorization response (redirect):

```
HTTP/1.1 302 Found
Location: https://tpp.fintech.cz/cb?code=a79819fe-4ff8-42cb-acd2-d176633df10b&state=af0ifjsldkj
```

Token request:

```
POST /token HTTP/1.1
host: as.jtbank.cz
Content-Type: application/x-www-form-urlencoded
Authorization: Basic dGVzdGNsaWVudGlkOnRlc3RjbGllbnRzZWNYZXQ=

grant_type=authorization_code
&code=a79819fe-4ff8-42cb-acd2-d176633df10b
&redirect_uri=https://tpp.fintech.cz/cb
```

Token response with binding access token to signId based on custom scope:

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store

{
  "access_token": "edfe58be-69c2-4863-97ba-11bb18004156",
  "refresh_token": "d304abc4-455e-4ca1-a754-1ff9cafe90e9",
  "token_type": "Bearer",
  "expires": 3600,
  "scope": "PISP signId7c44e312-5019-429a-841f-d9800be8688d"
}
```

Step III. Payment authorization initiation – specific for each bank (PUT /my/payments/{paymentId}/sign/{signId}) request:

```
PUT /my/payments/db8f4657-d874-4926-b8ff-3f521fea670c/sign/7c44e312-5019-429a-841f-d9800be8688d
Content-Type: application/json
X-Request-ID: 0d8c3226-53b9-4b8e-920f-5f836f1b5210
User-Involved: true
TPP-Name: api2bank
Authorization: Bearer edfe58be-69c2-4863-97ba-11bb18004156
{
  "authorizationType": "RAR"
}
```

Step III. Payment authorization initiation – specific for each bank (PUT /my/payments/{paymentId}/sign/{signId}) response:

```
HTTP/1.1 200
Content-Type: application/json
X-Request-ID: 0d8c3226-53b9-4b8e-920f-5f836f1b5210
{
  "transactionIdentification": "db8f4657-d874-4926-b8ff-3f521fea670c",
  "serviceLevel": {
    "code": "DOMESTIC"
  },
  "signInfo": {
    "state": "DONE",
    "signId": "7c44e312-5019-429a-841f-d9800be8688d"
  },
  "instructionStatus": "ACSP"
}
```

Example of expected but not required extension of introspect endpoint for ASPSP internal use – request:

```
POST /introspect token HTTP/1.1
host: as.jtbank.cz
Content-Type: application/x-www-form-urlencoded
Authorization: Basic dGVzdGNsaWVudGlkOnRlc3RjbGllbnRzZWNYZXQ=

token=7a877d61-3f77-3c65-a873-d182113b911f
```

Example of expected but not required extension of introspect endpoint for ASPSP internal use – response:

```
HTTP/1.1 200 OK
```

```

Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{
  "active":true,
  "username":1234567980,
  "sub":1234567980,
  "client_id":"4e6c2ae9-9954-42d9-b81d-46305b986cec",
  "iss": "https://asb.jtbank.cz",
  "exp": 1676414147,
  "iat": 1676410190,
  "token_type":"Bearer",
  "scope":"PISP signId7c44e312-5019-429a-841f-d9800be8688d",

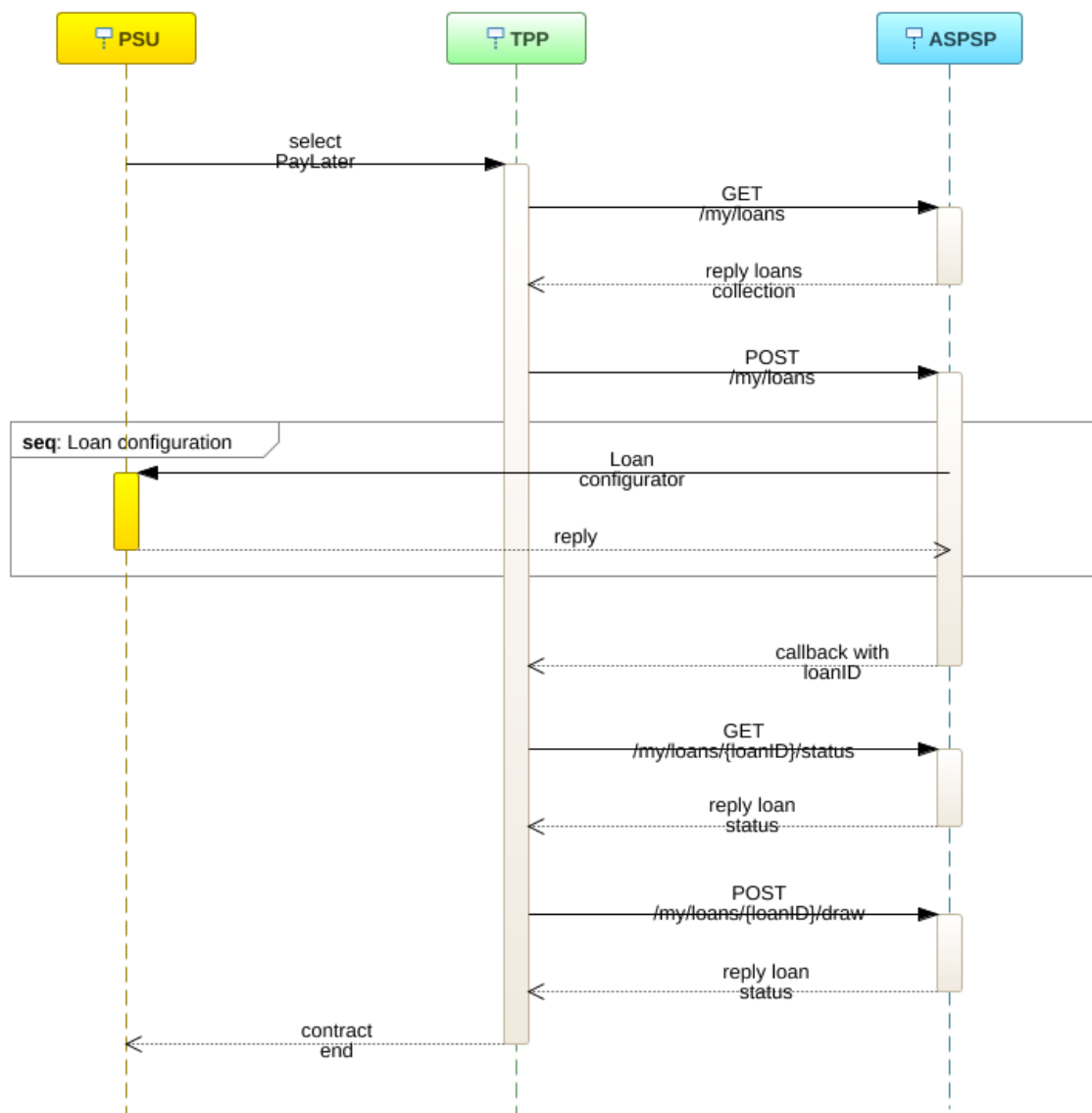
  "authorization_details":"U2FsdGVkX1/xaq5hQD/PU6JHEncGMTkqtrWwyyzD0JES
LsVIsNlmlf0zYYeJBHhg3ghWI3BuSm9eZeCYrMbj2VkmPW7rEFX0mI92bb7EZdQe9rmYf
CO5nI+1Q7jow38seZAGe2xoaXbrf47YaDZTNdUfFNkHQoEQaa1CjuhXuj3/bvpcj3dtTz
j4KEExI/alod6LmiiPP5xch/P0E3xGraclZW7QNMmqFSzCuZT7vWyHNAX9yKW8pPb+M0Nu
dCCSvpdCE4hsjijhtIu0PqI0oNw75wEkYdt2UrWN2pB2aOrB/hZiuYnkjrABmeerAfit7
pK4mLE9xr9KVqSpnMI/TFcmY/SUeAE1UBr1SI371GFbSuIPCaDk7KEYJeqJL9AibilMNH
5jlniRrNt8gaHf6DmaInOsibLwRM/s4Xprqgnm5PzoEEeWYu8Km05Z/kDtLLA9yHRFiA0
oDSScd7bnsnRtSx4cTX1I1T3qIop8HfQFd/41hWZfStQFCpC53Ry/fPGfRazkJ5fg8M+R
IRd/bOSfDJovF6cg/y1SQUW1FRQzBXwuPTxYT6kaSaoAaZeuhnMeQyX25wmDBDLrUsVF
jM0z3eTQWwnQa3jHpmWRlg2K1U9Rq6G+cBFQcUuq4Y6/r+aAyhjDE2JMk74SLCDqm3W7j
TjMxkrq8KekIR8vhP1qxZcFFSmU/cIoIcKkKUGsT+pMRtBjkIoWXT2ievQ0RHTQhQ8Z1
NjHYVkrNMdcEB3Ckuf36890lgjC+RXEd2/85SBGYLAMiemq/wuWlRMVmzeFoeY9GkLm7Z
KwUuTzKm3Y1+c7Jq/9NJ0Hen81FEec8GUR+23dXZdbwbPfnSfgtnQvYYBaMm0w4vrreKX
dykKYGG300L55mp4NaaNkJ4L5nmrUUVf5fM5Y49KV66fqV4rVnnaQOR0seuyWQz5HoLV
M4ePxKz4xEPZztbanu+R8x6/lg5q839o6o56K9Um0nWVcP88UkqAULeQjabr8R89pcAsk
7+SpGRqGLlgqeX+NasjsAO7oF5YrKBRqme1uwlFOP5KeHOJpAwoq5CTuRyJ7Gd2MjON4S
WRk7vpEueJZYAVyIHabQZ6mBXcnspmIngqifD746JFC+kCu4QaeHEKM1MJyUvMRC1zNJ/
QLbcq6mFjk53HIMH6uibP3TyYg=="
}

```

3.2.31 PayLater payment (loan)

This is a description of the solution for requesting PayLater payment in PSD2 world. Aim of this API is to offer to end users convenient way how to pay for goods and services by personalized loan from their own bank with possibility of choice of the bank right in the process of purchasing.

3.2.32 PayLater payment (loan) sequence diagram



3.2.33 Overview of unsecured loans initiated via TPP (GET/my/loans/{clientID})

This resource provides TPP (client) and end user with overview of unsecured loans initiated by end user in the past via particular TPP.

Resource characteristics

Mandatory:	yes
URI:	/my/loans/{clientID}
HTTP Method:	GET
Content-Type:	application/json

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: request **requires** the use of the third-party qualified certificate

Paging: no

Sorting: no

Filtering: no

Input character set: no restrictions in relation to chapter 3.2.1

Path parameters of the request:

PARAMETR	TYPE	MANDATORY	PURPOSE
clientID	text	No	TPP identifier for identification of particular TPP for which loans collection initiated by end user will be retrieved. TPP will be automatically identified by used certificate. This ClientID is needed to use when this resource is used outside of the PSD2 world without using PSD2certificates

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is always mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of GET response for calling, please see Chapter xxxx RESPONSE MESSAGE ELEMENTS for Overview of unsecured loans.

Error codes for GET Overview of unsecured loans:

HTTP STATUS CODE	ERROR CODE	PURPOSE
------------------	------------	---------

00	INVALID_REQUEST	Invalid request. A path parameter is in an unsuitable / invalid format
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated. The request was denied due to an invalid or missing bearer access token
401	UNAUTHORISED	Invalid/missing certificate = client is not authenticated. The request was denied due to an invalid or missing certificate
403	FORBIDDEN	Authentication with an expired access token or bearer access token having insufficient privileges
403	FORBIDDEN	Authentication with an expired TPP certificate having insufficient privileges

3.2.33.1 RESPONSE MESSAGE ELEMENTS for Overview of unsecured loans

LEVEL	MESSAGE ELEMENT	OCCURENCE	FORMAT TYPE	PRESENTATION
+	TPPName	[0..1]	Text	TPP name for which user has initiated loans in the past
+	Loans	[1..n]	Array of loans	Collection of loans
++	loanApplicationID	[1..1]	Text	Loan application identifier which serves as loan identifier
++	Amount	[1..1]	±	Amount of loan (price of goods or service)
+++	Value	[1..1]	Amount	Amount of loan (price of goods or service)
+++	Currency	[1..1]	CurrencyCode ISO4217	Currency of loan
++	From	[0..1]	Date	Date of drawing of loan
++	status	[1..1]	Text	Status one of: <ul style="list-style-type: none"> • IN APPROVAL PROCESS - loan application under approval • APPROVED – loan application approved • REJECTED - loan application rejected • DRAWN – loan drawn after

				application approved • DELETED - loan application deleted
--	--	--	--	--

3.2.34 Loan offers (GET/my/loanoffers{?loanPeriod,amount,currency})

This resource provides an end user with an information about loan offers from all other banks to make best decision for him/her.

Recommendation for banks or IDPs of banks, in case this resource is called with an access token and loan's parameters can be personalized according to end user's creditworthiness, is to put clear information on consent screen that loan's personalized parameters will be transferred to the TPP.

Resource characteristics

Mandatory:	no
URI:	/my/loanoffers{?loanPeriod,amount,currency}
HTTP Method:	GET
Authorization:	not required
Use certificate:	request requires the use of the third-party qualified certificate
Paging:	no
Sorting:	no
Filtering:	no
Filtering:	no

Input character set: no restrictions in relation to chapter 3.2.1

Query parameters of the request:

PARAMETR	TYPE	MANDATORY	PURPOSE
loanPeriod	text	no	Number of desired months for repayment of the loan by end user
amount	amount	yes	Amount of loan (price of goods or service)
currency	text	no	Currency of loan (ISO4217)

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is optional for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of GET response for calling, please see Chapter xxxx RESPONSE MESSAGE ELEMENTS for Loan offers.

Error codes for GET Loan offers:

<i>HTTP STATUS CODE</i>	<i>ERROR CODE</i>	<i>PURPOSE</i>
400	INVALID_REQUEST	Invalid request. A query parameter is in an unsuitable / invalid format
401	UNAUTHORISED	Invalid/missing certificate = client is not authenticated. The request was denied due to an invalid or missing certificate
403	FORBIDDEN	Authentication with an expired TPP certificate having insufficient privileges

3.2.34.1 RESPONSE MESSAGE ELEMENTS for Loan offers

<i>LEVEL</i>	<i>MESSAGE ELEMENT</i>	<i>OCCURENCE</i>	<i>FORMAT TYPE</i>	<i>PRESENTATION</i>
--------------	------------------------	------------------	--------------------	---------------------

+	loanExample	[1..n]	±	In case not authorized API call - Loan example for provided amount (price of goods or service) with all important parameters to make decision which loan offer is the best for end user. This is not meant as a showcase of best case scenario loan example for the most creditworthy clients but to show loan example for most common cases of loan provided to an average or median creditworthy client. It is necessary and strongly recommended to comply not to use this loan example as a showcase of best case scenario loan. Otherwise this whole resource is pointless for end user.
++	Amount	[1..1]	±	Amount of loan (price of goods or service)
+++	Value	[1..1]	Amount	Amount of loan (price of goods or service)
+++	Currency	[1..1]	CurrencyCode ISO4217	Currency of loan
++	interestRate	[1..1]	Text	Interest rate of loan in %
++	Apr	[1..1]	Text	Annual percentage rate of loan in %
++	monthlyPayment	[1..1]	Amount	Amount of monthly payment
++	totalPaments	[1..1]	Text	Number of total monthly payments
++	totalFee	[1..1]	Amount	Amount of fee for loan processing
++	totalRepaymentAmount	[1..1]	Amount	Total repayment amount of loan

3.2.35 Loan application (POST/my/loans)

This resource provides an option to submit loan application to the bank. End user is redirected to the bank side with loan configuration and other information needed for scoring. After loan application is complete, fully configured and authorized by end user on the bank side, bank sets up loan application and calls bank to the TPP informing about loan application ID and status. Loan application ID is always created regardless the loan application is approved or rejected.

Resource characteristics

Mandatory:	yes
URI:	/my/loans
HTTP Method:	POST
Authorization:	request requires the authorization of user/client as part of API calling
Use certificate:	request requires the use of the third-party qualified certificate
Paging:	no
Sorting:	no
Filtering:	no

Input character set: no restrictions in relation to chapter 3.2.1

Path parameters of the request:

PARAMETR	TYPE	MANDATORY	PURPOSE
-	-	-	-

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is always mandatory for this resource

3.2.35.1 REQUEST MESSAGE ELEMENTS for POST Loan application

LEVEL	MESSAGE ELEMENT	OCCURENCE	FORMAT TYPE	PRESENTATION
+	loanPeriod	[0..1]	Text	Number of desired months for repayment of the loan by end user
+	Amount	[1..1]	±	Amount of loan (price of goods or service)
++	Value	[1..1]	Amount	Amount of loan (price of goods or service)
++	Currency	[1..1]	CurrencyCode ISO4217	Currency of loan
+	Account	[1..1]	IBAN	Account of TPP on which amount of loan will be sent in when drawing the loan
+	callbackURL	[1..1]	text	URL link or package of loan application ID call back address.

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of POST response for calling, please see Chapter xxxx Callback for Loan application.

Error codes for POST Loan application:

HTTP STATUS CODE	ERROR CODE	PURPOSE
400	INVALID_REQUEST	Invalid request. A mandatory field is missing in the request, or it is in an unsuitable / invalid format
400	INVALID_ACCOUNT	Account number is in invalid format
400	INVALID_AMOUNT	Wrong amount. For instance, too low or high amount or wrong number format according to the number of decimal places according to the ISO 4217
400	INVALID_CURRENCY	The request contains a currency not trade/not supported

400	INVALIDD_LOAN_PERIOD	Wrong loan period - too low or high number of months
401	UNAUTHORISED	Invalid/missing certificate = client is not authenticated. The request was denied due to an invalid or missing certificate
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated. The request was denied due to an invalid or missing bearer access token
403	FORBIDDEN	Authentication with an expired TPP certificate having insufficient privileges
403	FORBIDDEN	Authentication with an expired access token or bearer access token having insufficient privileges

3.2.35.2 Callback for Loan application

LEVEL	MESSAGE ELEMENT	OCCURENCE	FORMAT TYPE	PRESENTATION
+	loanApplicationID	[1..1]	Text	ID of configured loan application
+	Status	[1..1]	Text	Status one of: <ul style="list-style-type: none"> • IN APPROVAL PROCESS - loan application under approval • APPROVED – loan application approved • REJECTED - loan application rejected

3.2.36 Loan application status (GET/my/loans/{loanApplicationID}/status)

This resource provides TPP with an information in which status is a loan application in order to know when to start drawing the loan in case of approved loan application or with an information that particular load application was approved and loan already drawn.

Resource characteristics

Mandatory: yes

URI: /my/loans/{loanApplicationID}/status

HTTP Method: GET

Authorization: not required

Use certificate: request **requires** the use of the third-party qualified certificate

Paging: no

Sorting: no

Filtering: no

Input character set: no restrictions in relation to chapter 3.2.1

Request periodicity: every 500 milliseconds

Path parameters of the request:

PARAMETR	TYPE	MANDATORY	PURPOSE
loanApplicationID	Text	Yes	Loan application identifier

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is optional for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Response headers](#).

The content of GET response for calling, please see Chapter xxxx RESPONSE MESSAGE ELEMENTS for Loan application status.

Error codes for GET Loan application status:

HTTP STATUS CODE	ERROR CODE	PURPOSE
------------------	------------	---------

400	INVALID_REQUEST	Invalid request. A mandatory path parameter is missing in the request, or it is in an unsuitable / invalid format
401	UNAUTHORISED	Invalid/missing certificate = client is not authenticated. The request was denied due to an invalid or missing certificate
403	FORBIDDEN	Authentication with an expired TPP certificate having insufficient privileges
404	ID_NOT_FOUND	Unknown loanApplicationID

3.2.36.1 RESPONSE MESSAGE ELEMENTS for Loan application status

LEVEL	MESSAGE ELEMENT	OCCURENCE	FORMAT TYPE	PRESENTATION
+	loanApplicationID	[1..1]	±	Loan application identifier which can serve as loan identifier in case of approved loan application and drawn loan
+	from	[0..1]	Date	Date of drawing of loan
+	amount	[1..1]	±	Amount of loan (price of goods or service)
++	value	[1..1]	Amount	Amount of loan (price of goods or service)
++	currency	[1..1]	CurrencyCode ISO4217	Currency of loan
+	status	[1..1]	Text	Status one of: <ul style="list-style-type: none"> • IN APPROVAL PROCESS - loan application under approval • APPROVED – loan application approved • REJECTED - loan application rejected • DRAWN – loan drawn after application approved • DELETED - loan application

				deleted
+	account	[0..1]	IBAN	Account of TPP on which amount of loan will be or was sent in when drawing the loan

3.2.37 Initiation of drawing a loan (POST /my/loans/{loanApplicationID}/draw)

This resource enables a TPP to initiate drawing of approved loan application. This resource is unauthorized and serves a TPP to receive financial amount, as end user already authorized loan application, in order to finish purchasing process.

Resource characteristics

Mandatory: yes

URI: /my/loans/{loanApplicationID}/draw

HTTP Method: POST

Authorization: not required

Use certificate: request **requires** the use of the third-party qualified certificate

Paging: no

Sorting: no

Filtering: no

Input character set: no restrictions in relation to chapter 3.2.1

Path parameters of the request:

PARAMETR	TYPE	MANDATORY	PURPOSE
loanApplicationID	text	Yes	Loan application identifier

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is optional for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of POST response for calling, please see Chapter xxxx RESPONSE for Initiation of drawing a loan.

Error codes for POST Initiation of drawing a loan:

<i>HTTP STATUS CODE</i>	<i>ERROR CODE</i>	<i>PURPOSE</i>
400	INVALID_REQUEST	Invalid request. A mandatory path parameter is missing in the request, or it is in an unsuitable / invalid format
401	UNAUTHORISED	Invalid/missing certificate = client is not authenticated. The request was denied due to an invalid or missing certificate
403	FORBIDDEN	Authentication with an expired TPP certificate having insufficient privileges
404	ID_NOT_FOUND	Unknown loan_applicationID

3.2.37.1 RESPONSE

Status 200

3.2.38 Loan application cancellation (DELETE/my/loans/{loanApplicationID})

This resource provides a TPP and an end user with an option to cancel still not approved loan application.

If a loan application is already approved it is not possible to cancel it by this resource. End user needs to cancel it with the particular bank itself.

Resource characteristics

Mandatory:	yes
URI:	/my/loans/{loanApplicationID}
HTTP Method:	DELETE
Authorization:	request requires the authorization of user/client as part of API calling
Use certificate:	request requires the use of the third-party qualified certificate
Paging:	no
Sorting:	no
Filtering:	no
Input character set:	no restrictions in relation to chapter 3.2.1

Path parameters of the request:

PARAMETR	TYPE	MANDATORY	PURPOSE
loan_applicationID	text	Yes	Loan application identifier

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is always mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of DELETE response for calling, please see Chapter xxxx RESPONSE for Loan application cancellation.

Error codes for DELETE Loan application cancellation:

HTTP STATUS CODE	ERROR CODE	PURPOSE
400	INVALID_REQUEST	Invalid request. A mandatory path parameter is missing in the request, or it is in an unsuitable / invalid format
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated. The request was denied due to an invalid or missing bearer access token
401	UNAUTHORISED	Invalid/missing certificate = client is not authenticated. The request was denied due to an invalid or missing certificate
403	FORBIDDEN	Authentication with an expired access token or bearer access token having insufficient privileges

3.2.39 Domestic direct debit consents – consent with payments initiated by the creditor

Overview of resources for the establishment, authorization and administration of domestic direct debit consent.

3.2.40 Domestic direct debit consent - instruction initiation (POST /my/accounts/{id}/directdebits)

Resource for entering a new domestic direct debit consent. The standard describes endpoints intended for establishing and managing direct debit authorizations. The description does not include services specified for the actual execution of an already authorized direct debit. The API only describes the domestic form of direct debit.

Resource characteristics

Mandatory: no

URI: /my/accaounts/{id}/direct-debits

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Direct%20Debits/-postDirectDebit>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: POST

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: no

Sorting: no

Filtration: no

Input character set: limitation is based on definitions from chapter 3.2.1

Example of request for new directdebit initialization:

```
POST https://api.bank.cz/v1/my/accounts/01fc4b99-359c-429e-8436-53ff14c082e9/directdebits
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa987785ac2
Date: Wed, 1 Jan 2021 09:25:01 GMT
User-Involved: false
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456

{
  "limit": {
    "directDebitPaymentLimit": {
      "value": 10000,
      "currency": "CZK"
    }
  },
  "directDebit": {
    "alias": "Platba za plyn",
    "execution": {
      "mode": "UNTIL_DATE",
      "interval": "MONTHLY",
      "valuePeriodLimit": 10000,
      "approvalDirectDebits": "N"
    },
    "validity": {
      "validFromDate": "2023-01-10",
      "validToDate": "2025-01-10",
      "maxIterations": 100
    }
  },
  "debtorAccount": {
    "identification": {
      "iban": "CZ0950518916425927369341"
    }
  }
},
```

```

"creditorAccount": {},
"remittanceInformation": {
  "unstructured": "/VS/7418529630/SS/1234567890",
  "structured": {
    "creditorReferenceInformation": {
      "reference": "string"
    }
  }
}
}

```

Example of response for new direct debit initialization:

```

HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 2258fffc-1587-44d4-9f2b-3aa9874d8854

{
  "directDebitIdentification": {
    "mandateIdentification": "a15bbb80-cbc6-475f-8483-3d21788e4eaf",
    "mandateRequestIdentification": "string"
  },
  "limit": {
    "directDebitPaymentLimit": {
      "value": 10000,
      "currency": "CZK"
    }
  },
  "directDebit": {
    "alias": "Platba za plyn",
    "execution": {
      "mode": "UNTIL_DATE",
      "interval": "MONTHLY",
      "valuePeriodLimit": 10000,
      "approvalDirectDebits": "N"
    },
    "validity": {
      "validFromDate": "2023-01-10",
      "validToDate": "2025-01-10",
      "maxIterations": 100
    }
  },
  "debtorAccount": {
    "identification": {
      "iban": "CZ0950518916425927369341"
    }
  }
}

```

```

},
"creditorAccount": {},
"remittanceInformation": {
  "unstructured": "/VS/7418529630/SS/1234567890",
  "structured": {
    "creditorReferenceInformation": {
      "reference": "string"
    }
  }
},
"signInfo": {
  "state": "OPEN",
  "signId": "164298331754922"
},
"instructionStatus": "string",
"statusChangeInfo": "string"
}

```

3.2.41 Domestic direct debit Consent detail (GET /my/accounts/{id}/directdebits/{mandateIdentification})

Resource for get detail about the domestic direct debit Consent.

It is at the choice of each bank whether to return details of any client's direct debit Consent or just those that the client has established through a particular TPP.

Resource characteristics

Mandatory: no

URI: /my/accounts/{id}/directdebits/{mandateIdentification}

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Direct%20Debits/-getDirectDebit>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: GET

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: no

Sorting: no

Filtration: no

Input character set: no restrictions in relation to chapter 3.1.1

Example of response for direct debit detail:

```
GET https://api.bank.cz/v1/my/accounts/01fc4b99-359c-429e-8436-53ff14c082e9/directdebits/a15bbb80-cbc6-475f-8483-3d21788e4eaf
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa987785ac2
Date: Wed, 1 Jan 2021 09:25:01 GMT
User-Involved: false
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456

{
  "directDebitIdentification": {
    "mandateIdentification": "a15bbb80-cbc6-475f-8483-3d21788e4eaf",
    "mandateRequestIdentification": "string"
  },
  "limit": {
    "directDebitPaymentLimit": {
      "value": 10000,
      "currency": "CZK"
    }
  },
  "directDebit": {
    "alias": "Platba za plyn",
    "execution": {
      "mode": "UNTIL_DATE",
      "interval": "MONTHLY",
      "valuePeriodLimit": 10000,
      "approvalDirectDebits": "N"
    },
    "validity": {
      "validFromDate": "2023-01-10",
      "validToDate": "2025-01-10",
      "maxIterations": 100
    }
  },
  "debtorAccount": {
    "identification": {
      "iban": "CZ0950518916425927369341"
    }
  },
  "creditorAccount": {},
  "remittanceInformation": {
    "unstructured": "/VS/7418529630/SS/1234567890",
    "structured": {
```

```

        "creditorReferenceInformation": {
            "reference": "string"
        }
    },
    "signInfo": {
        "state": "OPEN",
        "signId": "164298331754922"
    },
    "instructionStatus": "string",
    "statusChangeInfo": "string"
}

```

3.2.42 List of client's domestic direct debits Consents (GET /my/accounts/{id}/directdebits{?size,page,sort,order})

Paged list of client's domestic direct debits Consents. Each domestic direct debit Consent (Mandate) contains a unique id usable for URI referencing, e.g., for the domestic direct debit detail.

Additionally, the debtorAccount contains an API account id to call the account details resource to which the payment order is issued.

Resource characteristics

Mandatory: no

URI: /my/accounts/{id}/directdebits{?size,page,sort,order}

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Direct%20Debits/-getDirectDebits>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: GET

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: yes

Sorting: yes

Filtering: no

Input character set: no restrictions in relation to chapter 3.1.1

Example of response for list of direct debits:

```
GET https://api.bank.cz/v1/my/accounts/01fc4b99-359c-429e-8436-53ff14c082e9/directdebits/
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa987785ac2
Date: Wed, 1 Jan 2021 09:25:01 GMT
User-Involved: false
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456

{
  "pageNumber": 1,
  "pageCount": 12,
  "nextPage": 2,
  "pageSize": 25,
  "totalCount": 298,
  "directDebits": [
    {
      "directDebitIdentification": {
        "mandateIdentification": "a15bbb80-cbc6-475f-8483-3d21788e4eaf",
        "mandateRequestIdentification": "string"
      },
      "limit": {
        "directDebitPaymentLimit": {
          "value": 10000,
          "currency": "CZK"
        }
      },
      "directDebit": {
        "alias": "Platba za plyn",
        "execution": {
          "mode": "UNTIL_DATE",
          "interval": "MONTHLY",
          "valuePeriodLimit": 10000,
          "approvalDirectDebits": "N"
        },
        "validity": {
          "validFromDate": "2023-01-10",
          "validToDate": "2025-01-10",
          "maxIterations": 100
        }
      },
      "debtorAccount": {
        "identification": {
          "iban": "CZ0950518916425927369341"
        }
      }
    }
  ],
}
```



```

    "creditorAccount": {
      "identification": {
        "iban": "CZ0950518916425927369341"
      }
    },
    "remittanceInformation": {
      "unstructured": "/VS/7418529630/SS/1234567890",
      "structured": {
        "creditorReferenceInformation": {
          "reference": "string"
        }
      }
    },
    "signInfo": {
      "state": "OPEN",
      "signId": "164298331754922"
    },
    "instructionStatus": "string",
    "statusChangeInfo": "string"
  }
]
}

```

3.2.43 Change of existing Domestic direct debit Consent (PUT /my/accounts/{id}/directdebits/{id}/)

Resource for change of domestic direct debit Consent specification.

Resource characteristics

Mandatory: no

URI: /my/accounts/{id}/directdebits/{id}/

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Direct%20Debits/-putDirectDebit>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: PUT

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: no

Sorting: no

Filtration: no

Input character set: limitation is based on definitions from chapter 3.2.1

Example of request for change of direct debit:

```
PUT https://api.bank.cz/v1/my/accounts/01fc4b99-359c-429e-8436-53ff14c082e9/directdebits/a15bbb80-cbc6-475f-8483-3d21788e4eaf
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa987785ac2
Date: Wed, 1 Jan 2021 09:25:01 GMT
User-Involved: false
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456

{
  "directDebitIdentification": {
    "mandateIdentification": "a15bbb80-cbc6-475f-8483-3d21788e4eaf",
    "mandateRequestIdentification": "string"
  },
  "limit": {
    "directDebitPaymentLimit": {
      "value": 10000,
      "currency": "CZK"
    }
  },
  "directDebit": {
    "alias": "Platba za plyn",
    "execution": {
      "mode": "UNTIL_DATE",
      "interval": "MONTHLY",
      "valuePeriodLimit": 10000,
      "approvalDirectDebits": "N"
    },
    "validity": {
      "validFromDate": "2023-01-10",
      "validToDate": "2025-01-10",
      "maxIterations": 100
    }
  },
  "debtorAccount": {
    "identification": {
      "iban": "CZ0950518916425927369341"
    }
  },
  "creditorAccount": {},
  "remittanceInformation": {
    "unstructured": "/VS/7418529630/SS/1234567890",
    "structured": {
      "creditorReferenceInformation": {
```

```
        "reference": "string"
      }
    }
  }
}
```

Example of response for direct debit change request:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 2258fffc-1587-44d4-9f2b-3aa9874d8854

{
  "directDebitIdentification": {
    "mandateIdentification": "a15bbb80-cbc6-475f-8483-3d21788e4eaf",
    "mandateRequestIdentification": "string"
  },
  "limit": {
    "directDebitPaymentLimit": {
      "value": 10000,
      "currency": "CZK"
    }
  },
  "directDebit": {
    "alias": "Platba za plyn",
    "execution": {
      "mode": "UNTIL_DATE",
      "interval": "MONTHLY",
      "valuePeriodLimit": 10000,
      "approvalDirectDebits": "N"
    },
    "validity": {
      "validFromDate": "2023-01-10",
      "validToDate": "2025-01-10",
      "maxIterations": 100
    }
  },
  "debtorAccount": {
    "identification": {
      "iban": "CZ0950518916425927369341"
    }
  },
  "creditorAccount": {},
  "remittanceInformation": {
    "unstructured": "/VS/7418529630/SS/1234567890",
    "structured": {
      "creditorReferenceInformation": {
        "reference": "string"
      }
    }
  }
}
```

```

    }
  },
  "signInfo": {
    "state": "OPEN",
    "signId": "164298331754922"
  },
  "instructionStatus": "string",
  "statusChangeInfo": "string"
}

```

3.2.44 Domestic direct debit Consent delete (DELETE /my/accounts/{id}/directdebits/{id}/)

Resource for delete of domestic direct debit Consent specification.

Deleting of an authorised domestic direct debit Consent must be followed by the authorization process which is identical with the new domestic direct debit Consent authorisation process.

Resource characteristics

Mandatory: no

URI: /my/accounts/{id}/directdebits/{id}/

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Direct%20Debits/-deleteDirectDebit>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: DELETE

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: no

Sorting: no

Filtration: no

Input character set: limitation is based on definitions from chapter 3.2.1

Example of request for delete direct debit:

```

DELETE https://api.bank.cz/v1/my/accounts/01fc4b99-359c-429e-8436-53ff14c082e9/directdebits/a15bbb80-cbc6-475f-8483-3d21788e4eaf
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa987785ac2
Date: Wed, 1 Jan 2021 09:25:01 GMT
User-Involved: false

```

```
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456
```

Example of response for delete direct debit:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 2258fffc-1587-44d4-9f2b-3aa9874d8854

{
  "directDebitIdentification": {
    "mandateIdentification": "a15bbb80-cbc6-475f-8483-3d21788e4eaf",
    "mandateRequestIdentification": "string"
  },
  "signInfo": {
    "state": "OPEN",
    "signId": "164298331754922"
  },
  "instructionStatus": "string",
  "statusChangeInfo": "string"
}
```

In case of an unauthorised domestic direct debit, no content is returned in the response if the deleting was successful.

In case of an authorised domestic direct debit, the response contains a body (specified here below) with the payment identification and also signId, which is necessary to complete the deleting of the authorised payment process.

3.2.45 *Direct debit authorization Step I (GET my/accounts/{id}/directdebits/{mandateIdentification}/sign/{signId})*

First step of direct debit authorization.

Resource characteristics

Mandatory: no

URI: /my/accounts/{id}/directdebits/{mandateIdentification}/sign/{signId}

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Direct%20Debits/-getSignDirectDebit>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: GET

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: no

Sorting: no

Filtration: no

Input character set: no restrictions in relation to chapter 3.2.1

Query parameters of the request: **not defined**

Example of response:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 2258fffc-1587-44d4-9f2b-3aa9874d8854

{
  "scenarios": [
    [
      "CODE of The One of the methods ..."
    ],
    [
      "CODE of The second method ..."
    ]
  ],
  "signInfo": {
    "state": "OPEN",
    "signId": "164298331754922"
  }
}
```

3.2.46 Direct debit authorization Step II (POST /my/accounts/{id}/directdebits/{mandateIdentification}/sign/{signId})

Second step of direct debit authorization.

Resource characteristics

Mandatory: yes

URI: /my/accounts/{id}/directdebits/{mandateIdentification}/sign/{signId}

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Direct%20Debits/-postSignDirectDebit>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: POST

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: no

Sorting: no

Filtration: no

Input character set: no restrictions in relation to chapter 3.2.1

Example of request:

```
POST https://api.bank.cz/v1/my/accounts/01fc4b99-359c-429e-8436-53ff14c082e9/directdebits/a15bbb80-cbc6-475f-8483-3d21788e4eaf/sign/164298331754922
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa987785ac2
Date: Wed, 1 Jan 2021 09:25:01 GMT
User-Involved: false
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456

{
  "authorizationType": "SMS",
  "redirectUrl": "http://my.sometpp.com/..."
}
```

Example of response:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 2258fffc-1587-44d4-9f2b-3aa9874d8854

{
```

```

"authorizationType": "USERAGENT_REDIRECT",
"href": {
  "url": "http://www.bank.cz/authorization/164298331754922",
  "id": "164298331754922"
},
"method": "GET",
"formData": {
  "SAMLRequest": "45sa4d4e4e4fds5f6s4df4sd6f",
  "relayState": "token"
},
"rar": {
  "par_uri": "http://www.bank.cz/authorization/par",
  "scope": "signId164298331754922",
  "authorization_details":
"U2FsdGVkX1/xaq5hQD/PÜ6JHEncGMTkqtrWwyyzD0JESLsVIsNlmlf0zYYeJBHhg3ghW
I3BuSm9eZeCYrMbj2VkmPW7rEFX0mI92bb7EZdQe9rmYfCO5nI+1Q7jow38seZAGe2xoa
Xbrf47YaDZTNdUfFNkHQoEQaa1CjuhXuj3/bvpcj3dtTzj4KExI/alod6LmiiPP5xch/P
0E3xGraclZW7QMNMQfSzCuZT7vWyHNAX9yKW8pPb+M0NudCCSvpdCE4hsjijhtIu0PqI0
oNw75wEkYdt2UrWN2pB2aOrB/hZiuYnkjrABmeerAfit7pK4mLE9xr9KVqSpnMI/TFcmY
/SUeAE1UBr1SI371GFbSuIPCAdk7KEYJeqJL9AibilMNH5jlniRrNt8gaHf6DmaInOsiB
LwRM/s4Xprqgnm5PzoEEeWYu8Km05Z/kDtLLA9yHRFiA0oDSScD7bnsnRtSx4cTX1I1T3
qIop8HfQFd/4lhWZfStQFCpC53Ry/fPGfRazkJ5fg8M+RIRd/bOSfDJovF6cg/y1SQUW1
FRQzBXwuPTxYT6kaSaoAaZeuhnMeQyhX25wmDBDLrUsVFjM0z3eTQWwNqA3jHpmWRlg2K
lU9Rq6G+cBFQcUuq4Y6/r+aAyhjDE2JmK74SLCDqm3W7jTjMxkrq8KekIR8vhP1qxZcFF
SmU/cIolICkkKUGsT+pMRtBjkIoWXT2ievQ0RHTQhQ8Z1NjHYVkrNMDcEB3Ckuf36890l
gjc+RXEd2/85SBGYLAMiemq/wuWlRMVmzeFoeY9GkLm7ZKwUuTzKm3Y1+c7Jq/9NJ0Hen
81FEec8GUR+23dXZdbwbPfnSfgtnQvYYBaMm0w4vrreKXdykKYGG300L55mp4NaaNkJ4L
5nmrUUVf5fM5Y49KV66fqVk4rVnnaQORoseuyWQz5HoLVM4ePxKz4xEPZZtbanu+R8x6/
lg5q839o6o56K9Um0nWVcP88UkqAULeQjabr8R89pcAsk7+SpGRqGLlgqeX+NasjsAO7o
F5YrKBRqme1uwlFOP5KeHOJpAwoq5CTuRyJ7Gd2MjON4SWRk7vpEueJZYAVyIHabQZ6mB
XcnspmIngqiFD746JFC+kCu4QaeHEKM1MJyUvMRC1zNJ/QLbcq6mFjk53HIMH6uibP3Ty
Yg=="
  },
  "signInfo": {
    "state": "OPEN",
    "signId": "164298331754922"
  }
}

```

3.2.47 Direct debit authorization Step III (PUT /my/accounts/{id}/directdebits/{mandateIdentification}/sign/{signId})

Third step of direct debit authorization.

Resource characteristics

Mandatory: no

URI: /my/accounts/{id}/directdebits/{mandateIdentification}/sign/{signId}

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Direct%20Debits/-putSignDirectDebit>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: PUT

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: request **requires** the use of the qualified third-party certificate

Paging: no

Sorting: no

Filtration: no

Input character set: no restrictions in relation to chapter 3.2.1

Example of request:

```
PUT https://api.bank.cz/v1/my/accounts/01fc4b99-359c-429e-8436-53ff14c082e9/directdebits/a15bbb80-cbc6-475f-8483-3d21788e4eaf/sign/164298331754922
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa987785ac2
Date: Wed, 1 Jan 2021 09:25:01 GMT
User-Involved: false
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456
```

Example of response:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 2258fffc-1587-44d4-9f2b-3aa9874d8854

{
  "state": "DONE",
```

```
"pollInterval": 5000
}
```

3.3 API Balance Check

3.3.1 How to read API Balance Check

The sender of query for API account information must respect permitted characters. Otherwise, the failure to respect them may lead to rejection.

The defined list of enabled characters mainly applies to initiating a payment. For clarification, refer to each resource in the appropriate chapter. In any case, it is necessary to follow the definition in Chapter 1.2.5.

None of these elements may contain separate "/" (slash) at the beginning or end, or two successive slashes in the text. The permitted character set for payment initialisation is based only on the swift character set (that is, exclusively without diacritics - a different character set from the supported characters in CERTIS), i.e. they are the following characters:

```
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
```

The standard is based on the version of XML message caa.001.001.05 and also freely assumes some elements of the XML message pain.001.001.03 due to higher flexibility of application.

Only one query can be sent and processed per call.

We recommend not completing the account currency „debtorAccount.currency“. In the case of differences of the currency filled in pain.001 against the actual currency which specifies an account, you will avoid any possible complications in the payer bank, which may reject the entire message in the event of different values (the payment currency „transactionDetails.currency“ must be always specified).

All bank account numbers of bank clients in the Czech Republic are defined according to Decree No. 169/2011 Coll. on the rules for the creation of account numbers for making payments. The data element „debtorAccount.identification.iban“ requires an account number in the IBAN format which is defined by the international standard ISO 13616.

Structure of the table of elements:

- **LEVEL** – the character plus „+“ determines the level of element embedding. The basic level is marked „+“ and each another is marked with another character, for instance, the level two is „++“
- **MESSAGE ELEMENT** –element name in the camelCase format
- **OCCURRENCE** – describes the occurrence of fields:
 - [1..1] element is mandatory and only occurs once. In the case of a parent element, at least one embedded element must be filled in. If a mandatory element is embedded, the obligation does not pass to the parent element.
 - [0..1] element is optional and occurs only once.
 - [1..n] element is mandatory and occurs n-times. If the value "n" is not numerically defined, the number of repetitions is unlimited.
 - [0..n] element is not mandatory and occurs n-times. If the value "n" is not numerically defined, the number of repetitions is unlimited.
 - [0..0] element is not mandatory and in the specified type (request, attribute or element) does not have to occur, i.e. such element does not have to be implemented by a bank at all.
- **PAYMENT TYPE** – defines for which message types the current element is relevant
- **FORMAT TYPE** – defines the data format. It may be determined by the ISO 20022 standard valid for the item type PAIN.001 or CAAA.001 or another standard specifying the data structure (e.g., changes, dates, etc). Some items have a specific required format resulting from the payment system environment in the Czech Republic.
- **PRESENTATION** – contains a general role description

The format type defined as „±“ means the parent element which is then subdivided into other elements.

Conditions for element presence:

- When there is no OR but the parent element has more than 2 levels of elements, elements may be filled in cumulatively
- Parent element after [1..1] – at least one embedded element must be filled in
- Other see Sample description in the standard for XML pro pain.001, camt.053

3.3.2 List of API Balance Check resources

Specification of API Balance Check contains a description of resources for the acquisition of information on sufficient funds of the payer (client) at a specific financial institution.

An overview of resources:

- POST query for Balance Check

3.3.3 Query for Balance Check (POST /accounts/balanceCheck)

This is a resource for sending a query for Balance Check in a specific payment account of the account payer. The resource is not authorized directly by the account holder, through the authorization resource. Access to information must be granted by the client outside the interaction of this API before the resource is used.

A description of the process to grant access to the information is not part of this specification.

Resource characteristics

Mandatory: yes

URI: /accounts/balanceCheck

OpenAPI specification: <https://czech-ba.github.io/COBS/#/Balance%20check/-postAccountsBalanceCheck>

Technical repository: <https://github.com/Czech-BA/COBS>

HTTP Method: POST

Authorization: request **does not require** the authorization of user/client as part of the API calling

Use certificate: request **requires** the use of the third-party qualified certificate

Paging: no

Sorting: no

Filtering: no

Input character set: limitation is based on definitions from chapter 3.3.1

Query parameters of the request: **not defined**

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is **optional** for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of POST request and response for calling, please see Chapter 3.3.3.1 BASIC MESSAGE ELEMENTS Query for Balance Check

Return codes for the parameter „response“:

<i>CODE</i>	<i>DESCRIPTION</i>
APPR	Enough funds on this account
DECL	Unsufficient funds on this account

Error codes defined for the POST service Query for Balance Check:

<i>HTTP STATUS CODE</i>	<i>ERROR CODE</i>	<i>PURPOSE</i>
401	UNAUTHORISED	Missing certificate.
403	FORBIDDEN	Calling of the method which does not correspond to the licence, or invalid certificate.
400	FIELD_MISSING	Missing mandatory field in the request.
400	FIELD_INVALID	FIELD value is not valid.
400	AC02	[InvalidDebtorAccountNumber] – invalid account identifier in the request content.
400	AC09	[InvalidAccountCurrency] – invalid currency of the required account.
400	AC12	[InvalidAccountType] - account type does not match allowed account types (e.g., a non-paying account).
403	AG01	[TransactionForbidden] – absent consent to access to Balance Check at the account.
400	AM11	[InvalidTransactionCurrency] – the request contains a currency not trade/not supported.
400	AM12	[InvalidAmount] – wrong amount. For instance, too low or high amount or wrong number format according to the number of decimal places according to the ISO 4217.

400	FF01	[Invalid File Format] – invalid JSON format or other technical problem with the query processing.
400, 50x	NARR	Narrative – a general reason for rejecting the payment, with an addition of error-related information.
400	RF01	[NotUniqueTransactionReference] – not unique request identifier.
400	RR10	[InvalidCharacterSet] – invalid character set in the request.

3.3.3.1 BASIC MESSAGE ELEMENTS Query for Balance Check

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
+	exchangeIdentification	[1..1]	CISP	Max18Text	Clear query identification
+	card	[0..1]	CISP	±	Transaction card
++	cardholderName	[0..1]	CISP	Max45Text	Card holder name
++	maskedPan	[1..1]	CISP	Max30Text	Masked card number
+	debtorAccount	[1..1]	CISP	±	Payer account
++	identification	[1..1]	CISP	±	Payer account identification
+++	iban	[1..1]	CISP	IBAN2007Identifier	IBAN
++	currency	[0..1]	CISP	CurrencyCode, ISO 4217	Payer account currency
+	authenticationMethod	[0..1]	CISP	CodeSet	Client verification method
+	merchant	[0..1]	CISP	±	Merchant executing the transaction
++	identification	[1..1]	CISP	Max35Text	Merchant identification
++	type	[0..1]	CISP	Code	Merchant type
++	shortName	[1..1]	CISP	Max35Text	Merchant name

++	commonName	[1..1]	CISP	Max70Text	Merchant name as stated in the payment receipt
++	address	[0..1]	CISP	Max140Text	Merchant address
++	countryCode	[0..1]	CISP	CountryCode, ISO 3166 (2 alphanumeric characters code version)	Merchant country
++	merchantCategoryCode	[1..1]	CISP	Min3Max4Text, ISO 18245	Merchant code following the transaction type
+	transactionDetails	[1..1]	CISP	±	Transaction details
++	currency	[1..1]	CISP	CurrencyCode, ISO 4217	Balance query currency
++	totalAmount	[1..1]	CISP	Max18.5Amount	Balance query amount

3.3.3.2 MESSAGE ELEMENTS Response for Balance Check

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+	responseIdentification	[1..1]	Number (integers)	Unique identification of response to query for Balance Check (from ASPSP).
+	exchangeIdentification	[1..1]	IntMax18Digits	Repeated identification of a payment transaction (query for Balance Check) from the issuer of the card to which the request for Balance Check linked to the account.
+	response	[1..1]	Code set	Result code of query for Balance Check.

Return codes for the parameter „response“ – Code set:

CODE	DESCRIPTION
APPR	Enough funds on this account
DECL	Unsufficient funds on this account

3.4 Know your customer and User info service

Definition of optional APIs designed to pass information about the authenticated user (bank client). The chapter defines two resources. One for the transfer of so-called user info, ie contact information about the client. The second, the so-called profile, allows you to pass on client information obtained through AML-based physical verification.

3.4.1 User info service

Retrieves the consented UserInfo and other informations about the logged-in subject (end-user).

Resource characteristics

Mandatory: no

URI: /userinfo

Technical repository: <https://github.com/Czech-BA/BankID>

HTTP Method: GET

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: no

Paging: no

Sorting: no

Filtering: no

Scopes: yes

Input character set: no restrictions in relation to chapter 3.1.1

Query parameters of the request:

PARAMETR	TYPE	MANDATORY	PURPOSE
-	-	-	-

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is always mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of GET response for calling, please see Chapter 3.4.1.1 RESPONSE MESSAGE ELEMENTS User info service

Error codes defined for the GET User info service:

HTTP STATUS CODE	ERROR CODE	PURPOSE
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated. The request was denied due to an invalid or missing bearer access token. Also used to indicate that the client (registration URI) doesn't exist on the server.
403	FORBIDDEN	Authentication with an expired access token or bearer access token having insufficient privileges

3.4.1.1 RESPONSE MESSAGE ELEMENTS User info service

LEVEL	MESSAGE ELEMENT	OCCURENCE	FORMAT TYPE	PRESENTATION
-------	-----------------	-----------	-------------	--------------

+	sub	[1..1]	Text	Identifier for the End-User at the Issuer.
+	name	[0..1]	Text	End-User's full name in displayable form including all name parts, possibly including titles and suffixes, ordered according to the End-User's locale and preferences.
+	given_name	[0..1]	Text	Given name(s) or first name(s) of the End-User. Note that in some cultures, people can have multiple given names; all can be present, with the names being separated by space characters.
+	family_name	[0..1]	Text	Surname(s) or last name(s) of the End-User. Note that in some cultures, people can have multiple family names or no family name; all can be present, with the names being separated by space characters.
+	middle_name	[0..1]	Text	Middle name(s) of the End-User. Note that in some cultures, people can have multiple middle names; all can be present, with the names being separated by space characters. Also note that in some cultures, middle names are not used.
+	nickname	[0..1]	Text	Casual name of the End-User that may or may not be the same as the given_name. For instance, a nickname value of Mike might be returned alongside a given_name value of Michael.
+	preferred_username	[0..1]	Text	Shorthand name by which the End-User wishes to be referred to at the RP, such as janedoe or j.doe. This value MAY be any valid JSON string including special characters such as @, /,

				or whitespace. The API consumer MUST NOT rely upon this value being unique.
+	title_prefix	[0..1]	Text	User's title prefix
+	title_suffix	[0..1]	Text	User`s title suffix
+	email	[0..1]	Text	End-User's preferred e-mail address. Its value MUST conform to the RFC 5322 [RFC5322] addr-spec syntax. The API consumer MUST NOT rely upon this value being unique.
+	email_verified	[0..1]	Boolean	True if the End-User's e-mail address has been verified; otherwise false. When this Claim Value is true, this means that the OP took affirmative steps to ensure that this e-mail address was controlled by the End-User at the time the verification was performed. The means by which an e-mail address is verified is context-specific, and dependent upon the trust framework or contractual agreements within which the parties are operating.
+	gender	[0..1]	Text	End-User's gender. Values defined by this specification are female and male. Other values MAY be used when neither of the defined values are applicable.
+	birthdate	[0..1]	Text	End-User's birthday, represented as an ISO 8601:2004 [ISO8601-2004] YYYY-MM-DD format. The year MAY be 0000, indicating that it is omitted. To represent only the year, YYYY format is allowed. Note that depending

				on the underlying platform's date related function, providing just year can result in varying month and day, so the implementers need to take this factor into account to correctly process the dates.
+	zoneinfo	[0..1]	Text	String from zoneinfo [zoneinfo] time zone database representing the End-User's time zone. For example, Europe/Paris or America/Los_Angeles.
+	locale	[0..1]	Text	End-User's locale, represented as a BCP47 [RFC5646] language tag. This is typically an ISO 639-1 Alpha-2 [ISO639-1] language code in lowercase and an ISO 3166-1 Alpha-2 [ISO3166-1] country code in uppercase, separated by a dash. For example, en-US or fr-CA. As a compatibility note, some implementations have used an underscore as the separator rather than a dash, for example, en_US; Relying Parties MAY choose to accept this locale syntax as well.
+	phone_number	[0..1]	Text	End-User's preferred telephone number. E.164 [E.164] is RECOMMENDED as the format of this Claim, for example, +1 (425) 555-1212 or +56 (2) 687 2400. If the phone number contains an extension, it is RECOMMENDED that the extension be represented using the RFC 3966 [RFC3966] extension syntax, for example, +1 (604) 555-1234;ext=5678.
+	phone_number_verified	[0..1]	Boolean	True if the End-User's phone number has been verified; otherwise false. When this Claim Value is true, this means

				that the OP took affirmative steps to ensure that this phone number was controlled by the End-User at the time the verification was performed. The means by which a phone number is verified is context-specific, and dependent upon the trust framework or contractual agreements within which the parties are operating. When true, the phone_number Claim MUST be in E.164 format and any extensions MUST be represented in RFC 3966 format.
+	updated_at	[0..1]	Number	Time the End-User's information was last updated. Its value is a JSON number representing the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time.

3.4.1.2 SCOPES AVAILABLE for User info service

Defined scopes are used to parameterize the API. Scope selects applications consuming API and its use must be confirmed by the user on the so-called consent screen during authentication.

The set of scopes defined here should be included in the API definition on the bank's development panel.

The definition and use of scopes are based on the description in chapter 3.3 of RFC6749 [\[link\]](#).

Scopes definition for the userinfo service:

SCOPE NAME	RESPONSE MESSAGE ELEMENTS PROVIDED WITHIN SCOPE
profile.name	name, given_name, family_name, middle_name, nickname and preferred_username
profile.gender	gender
profile.email	email and email_verified

profile.phone_number	phone_number and phone_number_verified
profile.birthdate	birthdate
profile.zoneinfo	zoneinfo
profile.locale	locale
profile.updated_at	updated_at

3.4.2 Example of request and response for Query user info service

3.4.2.1.1 Example of request

Request headers:

```
GET https://api.bank.cz/v1/kyc/userinfo
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:21:01 GMT
API-key: 00000000-1212-0f0f-a0a0-123456789abc
Authorization: Bearer AbCdEf123456
```

3.4.2.1.2 Example of correct response

Response 200 OK Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 200 OK Body

```
{
  "sub": "23f1ac00-5d54-4169-a288-794ae2ead0c4",
  "name": "Jan Novák",
  "given_name": "Jan",
  "family_name": "Novák",
```

```
"middle_name": "",
"nickname": "Zorro",
"preferred_username": "Jan",
"email": "J.novak@email.com",
"email_verified": true,
"gender": "male",
"birthdate": "1970-08-01",
"zoneinfo": "Europe/Prague",
"locale": "cs_CZ",
"phone_number": "+420123456789",
"phone_number_verified": false,
  "updated_at": 1568188433000
}
```

3.4.2.2 Example of signed Request and Response:

If the bank would like to provide data in a signed form to ensure the integrity of the information, it is possible to use the JWS (JSON Web Signature) principle. In this case, the request contains the content type value in the form of application/jwt. The answer is therefore in the form of a signed JWT according to RFC7515 [\[link\]](#).

Request:

```
GET https://api.bank.cz/v1/kyc/userinfo
Content-Type: application/jwt
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:21:01 GMT
API-key: 00000000-1212-0f0f-a0a0-123456789abc
Authorization: Bearer AbCdEf123456
```

Response Header:

```
HTTP/1.1 200 OK
Content-Type: application/jwt
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Signed Response 200 OK Body

Response consists of three parts separated by a dot.

- The first part is the header, which determines, among other things, the cipher used. This is JSON content encoded with BASE64.
- The second part introduces JSON payload with content corresponding to the response content in case of JSON transmission. This part is also encoded into BASE64.
- The last part is a signed hash of the second part created by the algorithm described in the "header".

```
ewoidHlwIjoiVXNlcmluZm8iLAoiYWxnIjoiSFMyNTYiCn0K.  
ewoic3ViIjogIjIzZjFhYzAwLTVkbWNTQmNTNDE2OS1hMjg4LTc5NGF1MmVhZDBjNCIsCiJuYW  
1lIjogIkp3biBOb3bDoWsiLAoiZ212ZW5fbmFtZSI6ICJKYW4iLAogIAkiZmFtaWx5X25h  
bWUioiAiTm92w6FrIiwKICAJIm1pZGRsZV9uYW1lIjogIiIsCiAgCSJuaWNrbmFtZSI6IC  
Jab3JybyIsCiAgCSJwcmVmZXJyZWVfdXNlcmluZm8iLAoiSmFuIiwKIakiZW1haWwiOiAi  
Si5ub3Zha0BlbWVpbC5jb20iLAogIAkiZW1haWxfdmVyaWZpZWQioiB0cnVlLAogIAkiZ2  
VuZGVyIjogIm1hbGUiLAogIAkiYmlydGhkYXRlIjogIjE5NzAtMDgtMDEiLAogIAkiem9u  
ZWluZm8iOiAiRXVyb3BlL1ByYWdlZSI6ICAgCSJsb2NhbGUioiAiY3NfQ1oiLAogIAkicG  
hvbmVfbnVtYmVyIjogIi0MjAxMjM0NTY3ODkiLAogIAkicGhvbmVfbnVtYmVyX3Zlcm1m  
aWVvIjogZmFsc2UsCiAidXBkYXRlZGF9hdCI6IDE1Njg5ODg0MzZwMDAKfQo=  
cDEuITldHi14RUEGG3cxEWOXFzZWIG46JmMWZu+/vXImNBh0QUXlgq07xbYWYzF2biUl77  
+9dXZ2XAEjbFglfgEDKRXYrgUuUm8kWzNKHgQKIARNPhIBMHlbMtS9O0FmZR15I31w07Fd  
PC4ndUVkrBd6Dj4nWmEKe9aHHjFgAxFIUTI8bgFbPfk6lbcXLFZsWVZpKBw0MmlUPQFw
```

Header:

```
As JSON:  
{  
  "typ": "Userinfo",  
  "alg": "HS256"  
}  
And like BASE64 encoded string:  
ewoidHlwIjoiVXNlcmluZm8iLAoiYWxnIjoiSFMyNTYiCn0K
```

Payload:

```
As JSON:  
{  
  "sub": "23f1ac00-5d54-4169-a288-794ae2ead0c4",  
}
```



```
"name": "Jan Novák",
"given_name": "Jan",
"family_name": "Novák",
"middle_name": "",
"nickname": "Zorro",
"preferred_username": "Jan",
"email": "J.novak@email.com",
"email_verified": true,
"gender": "male",
"birthdate": "1970-08-01",
"zoneinfo": "Europe/Prague",
"locale": "cs_CZ",
"phone_number": "+420123456789",
"phone_number_verified": false,
"updated_at": 1568188433000
}
```

And like BASE64 encoded string:

```
ewoic3ViIjogIjIzZjFhYzAwLTVkNTQtNDE2OS1hMjg4LTc5NGFlMmVhZDBjNCIsCiJuYW
1lIjogIkphbiBOb3bDoWsiLAoiZ2l2ZW5fbmFtZSI6ICJKYW4iLAogIAkiZmFtaWx5X25h
bWUiOiAiTm92w6FrIiwKICAJIm1pZGRsZV9uYW1lIjogIiIsCiAgCSJuaWNrbmFtZSI6IC
Jab3JybyIsCiAgCSJwcmVmZXJyZWRfdXNlcm5hbWUiOiAiSmFuIiwKIakiZWlhaWwiOiAi
Si5ub3Zha0BlbWFpbC5jb20iLAogIAkiZWlhaWxmdmVyaWZpZWQiOiB0cnVlLAogIAkiZ2
VuZGVyIjogIm1hbGUiLAogIAkiYmlydGhkYXRlIjogIjE5NzAtMDgtMDEiLAogIAkiem9u
ZWluZm8iOiAiRXVyb3BlL1ByYWdlZSI6ICAgCSJsb2NhbGUiOiAiY3NfQ1oiLAogIAkicG
hvbmVfbnVtYmVyIjogIi0MjAxMjM0NTY3ODkiLAogIAkicGhvbmVfbnVtYmVyX3Zlcm1m
aWVkiIjogZmFsc2UsCiAidXBkYXRlZl9hdCI6IDE1Njg4ODg0MzZmMDAKfQo=
```

Signature:

```
cDEuITlDHi14RUEGG3cxEWoXFzZWIG46JmMWZu+/vXImNBh0QUXlgq07xbYWYzF2biU177
+9dXZ2XAEjbfG1fgEDKRXYrgUuCm8kWzNKHgQKIARNPhIBMHlbMtS9O0FmZR15I31w07Fd
PC4ndUVkRbD6Dj4nWmEKe9aHHjFgAxFIUTI8bgFbPfk6lbcXLFZsWVZpKBw0MmlUPQFw
```

3.4.3 User profile

Retrieves the consented user profile and client information obtained during the AML verification process about the logged-in subject (end-user).

Resource characteristics

Mandatory: no

URI: /profile

Technical repository: <https://github.com/Czech-BA/BankID>

HTTP Method: GET

Authorization: request **requires** the authorization of user/client as part of API calling

Use certificate: no

Paging: no

Sorting: no

Filtering: no

Scopes: yes

Input character set: no restrictions in relation to chapter 3.1.1

Query parameters of the request:

PARAMETR	TYPE	MANDATORY	PURPOSE
-	-	-	-

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.1 Request headers](#).

Conditional use of common header parameters:

- **Authorization** parameter is always mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter [1.5.2 Reponse headers](#).

The content of GET response for calling, please see Chapter 3.5.4.1 RESPONSE MESSAGE ELEMENTS User profile service

Error codes defined for the GET User profile service:

HTTP STATUS CODE	ERROR CODE	PURPOSE
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated. The request was denied due to an invalid or missing bearer access token. Also used to indicate that the client (registration URI) doesn't exist on the server.
403	FORBIDDEN	Authentication with an expired access token or bearer access token having insufficient privileges

3.4.3.1 RESPONSE MESSAGE ELEMENTS User profile service

LEVEL	MESSAGE ELEMENT	OCCURENCE	FORMAT TYPE	PRESENTATION
+	sub	[1..1]	Text	Identifier for the End-User at the Issuer.
+	title_prefix	[0..1]	Text	User's title prefix
+	title_suffix	[0..1]	Text	User`s title suffix
+	given_name	[0..1]	Text	Given name(s) or first name(s) of the End-User. Note that in some cultures, people can have multiple given names; all can be present, with the names being separated by space characters.
+	family_name	[0..1]	Text	Surname(s) or last name(s) of the End-User. Note that in some cultures, people can have multiple family names or no family name; all can be present, with the names being separated by space characters.
+	middle_name	[0..1]	Text	Middle name(s) of the End-User. Note that in some cultures, people can have multiple

				middle names; all can be present, with the names being separated by space characters. Also note that in some cultures, middle names are not used.
+	gender	[0..1]	Text	End-User's gender. Values defined by this specification are female and male. Other values MAY be used when neither of the defined values are applicable.
+	birthdate	[0..1]	Text	End-User's birthday, represented as an ISO 8601:2004 [ISO8601-2004] YYYY-MM-DD format. The year MAY be 0000, indicating that it is omitted. To represent only the year, YYYY format is allowed. Note that depending on the underlying platform's date related function, providing just year can result in varying month and day, so the implementers need to take this factor into account to correctly process the dates.
+	birthnumber	[0..1]	Text	User's birth number. Birth number is required if nationality is Czech ("CZ").
+	birthcountry	[0..1]	Text	User's birth country in ISO 3166-1 format, subtype ALPHA-2. This means two letters in uppercase.
+	birthplace	[0..1]	Text	User's birth place.
+	primary_nationality	[0..1]	Text	User's primary nationality, ISO 3166-1 format, subtype ALPHA-2. This means two letters in uppercase.
+	other_nationalities	[1..n]	Text	Country codes in ISO 3166-1 format subtype of ALPHA-2. This means two letters in uppercase.
+	maritalstatus	[0..1]	Text	Martial status. One of: <ul style="list-style-type: none"> • COHABITATION - cohabitation status. • MARRIED - married status • DIVORCED - divorced status • REGISTERED_PARTNERSHIP - registered partnership status • REGISTERED_PARTNERSHIP_CANCELED - registered partnership canceled status • WIDOWED - widowed status • SINGLE - single status

				<ul style="list-style-type: none"> UNKNOWN - unknown status
+	addresses	[1..n]	±	User's addresses
++	type	[1..1]	AddressTypeCode	Type of address element as an ENUM: <ul style="list-style-type: none"> PERMANENT_RESIDENCE - permanent residence address SECONDARY_RESIDENCE - secondary residence address UNKNOWN - unknown address type
++	street	[1..1]	Text	Street
++	buildingapartment	[1..1]	Text	Address land registry number.
++	streetnumber	[1..1]	Text	Additional address house number.
++	city	[1..1]	Text	City name.
++	zipcode	[1..1]	Text	Zip code of the address.
++	country	[1..1]	Text	Country Code, ISO 3166-1 format, subtype ALPHA-2. This means two letters in uppercase.
+	idcards	[1..n]	±	User's id cards
++	type	[1..1]	Text	Id card type code: <ul style="list-style-type: none"> IC - Identity card. P - Passport of the Czech Republic resident. DL - Driving license IP - International Passport (of another state than the Czech Republic) DP - Diplomatic passport LRP - Long-term residence permit PRP - Permanent residence permit SRP - Short-term residence permit GC - Green card UNKNOWN - Unknown id card type
++	description	[0..1]	Text	Localized id card type description.

++	country	[1..1]	Text	Country for which is id card valid in ISO 3166-1 format, subtype ALPHA-2. This means two letters in uppercase.
++	issuer	[0..1]	Text	Office which issued id card.
++	number	[1..1]	Text	Number of id card.
++	valid_to	[1..1]	Text	Id card validity.
+	paymentaccounts	[1..n]	Array of Text	User's payment account numbers in CZ IBAN format.
+	email	[0..1]	Text	End-User's preferred e-mail address. Its value MUST conform to the RFC 5322 [RFC5322] addr-spec syntax. The API consumer MUST NOT rely upon this value being unique.
+	phone_number	[0..1]	Text	End-User's preferred telephone number. E.164 [E.164] is RECOMMENDED as the format of this Claim, for example, +1 (425) 555-1212 or +56 (2) 687 2400. If the phone number contains an extension, it is RECOMMENDED that the extension be represented using the RFC 3966 [RFC3966] extension syntax, for example, +1 (604) 555-1234;ext=5678.
+	updated_at	[0..1]	Number	Time the End-User's information was last updated. Its value is a JSON number representing the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time.

3.4.3.2 SCOPES AVAILABLE for User profile service

Defined scopes are used to parameterize the API. Scope selects applications consuming API and its use must be confirmed by the user on the so-called consent screen during authentication.

The set of scopes defined here should be included in the API definition on the bank's development panel.

The definition and use of scopes are based on the description in chapter 3.3 of RFC6749 [\[link\]](#).

Scopes definition for the user profile service:

SCOPE NAME	RESPONSE MESSAGE ELEMENTS PROVIDED WITHIN SCOPE
profile.name	given_name, family_name, middle_name, nickname and preferred_username
profile.titles	title_prefix and title_suffix
profile.gender	gender
profile.birthdate	birthdate
profile.birthnumber	birthnumber
profile.birthcountryBirthplace	birthcountry and birthplace
profile.primary_nationality	primary_nationality
profile.other_nationalities	other_nationalities.country_code
profile.martialstatus	martialstatus
profile.addresses	addresses.type, addresses.street, addresses.buildingapartment, addresses.streetnumber, addresses.city, addresses.zipcode, addresses.country
profile.idcards	idcards.type, idcards.description, idcards.country, idcards.issuer, idcards.number, idcards.valid_to
profile.paymentaccounts	CZ IBAN format: prefix, number, bankcode, iban, bic
profile.email	email
profile.phone_number	phone_number
profile.updated_at	updated_at

3.4.4 Example of request and response for Query user profile service

3.4.4.1 Example of request

Request headers:

```
GET https://api.bank.cz/v1/kyc/profile
Content-Type: application/json
```

```
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:21:01 GMT
API-key: 00000000-1212-0f0f-a0a0-123456789abc
Authorization: Bearer AbCdEf123456
```

3.4.4.2 Example of correct response

Response **200 OK** Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response **200 OK** Body

```
{
  "sub": "23f1ac00-5d54-4169-a288-794ae2ead0c4",
  "given_name": "Jan",
  "family_name": "Novák",
  "middle_name": "",
  "gender": "male",
  "birthdate": "1970-08-01",
  "birthnumber": "700801/0147",
  "birthcountry": "CZ",
  "birthplace": "Praha 4",
  "primary_nationality": "CZ",
  "other_nationalities": [
    {
      "country_code": "DE"
    }
  ],
  "maritalstatus": "MARRIED",
  "email": "J.novak@email.com",
  "phone_number": "+420123456789",
  "addresses": [
```



```

    {
      "type": "PERMANENT_RESIDENCE",
      "street": "Olbrachtova",
      "buildingapartment": "1929",
      "streetnumber": "62",
      "city": "Praha",
      "zipcode": "14000",
      "country": "CZ"
    }
  ],
  "idcards": [
    {
      "type": "IC",
      "description": "Občanský průkaz",
      "country": "CZ",
      "issuer": "Úřad městské části Praha 4",
      "number": "123456789",
      "valid_to": "2023-10-11"
    }
  ],
  "paymentAccounts": [
    "CZ0708000000001019382023"
  ],
  "updated_at": 1568188433000
}

```

3.5 API Account Statement

3.5.1 How to read API Account Statement

The sender of query for API account information must respect permitted characters. Otherwise, the failure to respect them may lead to rejection.

The defined list of enabled characters mainly applies to initiating a payment. For clarification, refer to each resource in the appropriate chapter. In any case, it is necessary to follow the definition in Chapter 1.2.5.

Only one query can be sent and processed per call.

Structure of the table of elements:

- **LEVEL** – the character plus „+“ determines the level of element embedding. The basic level is marked „+“ and each another is marked with another character, for instance, the level two is „++“
- **MESSAGE ELEMENT** – element name in the camelCase format
- **OCCURRENCE** – describes the occurrence of fields:
 - [1..1] element is mandatory and only occurs once. In the case of a parent element, at least one embedded element must be filled in. If a mandatory element is embedded, the obligation does not pass to the parent element.
 - [0..1] element is optional and occurs only once.
 - [1..n] element is mandatory and occurs n-times. If the value "n" is not numerically defined, the number of repetitions is unlimited.
 - [0..n] element is not mandatory and occurs n-times. If the value "n" is not numerically defined, the number of repetitions is unlimited.
 - [0..0] element is not mandatory and in the specified type (request, attribute or element) does not have to occur, i.e. such element does not have to be implemented by a bank at all.
- **PAYMENT TYPE** – defines for which message types the current element is relevant
- **FORMAT TYPE** – defines the data format. It may be determined by the ISO 20022 standard valid for the item type PAIN.001 or CAAA.001 or another standard specifying the data structure (e.g., changes, dates, etc). Some items have a specific required format resulting from the payment system environment in the Czech Republic.
- **PRESENTATION** – contains a general role description

The format type defined as „±“ means the parent element which is then subdivided into other elements.

Conditions for element presence:

- When there is no OR but the parent element has more than 2 levels of elements, elements may be filled in cumulatively
- Parent element after [1..1] – at least one embedded element must be filled in

3.5.2 List of API resources Account Statement

Specification of API Account Statement contains a description of resources for the acquisition of information about the account statements of the bank client.

Overview of resources:

- GET list of available account statements

- GET download the statement file

3.5.3 List of available account statements (GET /my/accounts/{id}/statements{?year,month,availability })

List of client account statements.

Resource characteristics

Mandatory: yes

URI:

/my/accounts/{id}/statements{?year,halfyear,quarter,month,week,day,availability,page,size}
or
/my/accounts/{id}/statements{?from,to,matching,availability,page,size}

HTTP Method: GET

Authorization: request requires the authorization of user/client as part of API calling

Use certificate: request requires the use of the qualified third-party certificate

Paging: yes

Sorting: no (not available as API parameter, but implicit sorting should to be implemented to facilitate a predictable paging.

Filtering: yes

Input character set: no restrictions in relation to chapter 3.1.1

3.5.3.1 REQUEST STRUCTURE Available Account Statements

Query parameters of the request:

PARAMETER	TYPE	MANDATORY	PURPOSE
id	Text	Yes	API payment account identifier from the response to a query on the list of accounts.
year	Number	No	Year of account statement.
The following parameters must only be used in combination with the "year" parameter:			
halfyear	Number	No	Halfyear of account statement. values 1, 2.
quarter	Number	No	Quarter of account statement. Values (1, 2, 3, 4).
month	Number	No	Month of account statement.Values (1-12)
week	Number	No	Week of account statement. Denotes week of the year as number (1-53).

day	Number	No	Day of account statement in case of daily frequency. Values (1-366).
The following parameters can be used without previous parameters (year, quarter, month ...):			
from	Date	No	From date, start of a matching period
to	Date	No	End date of a matching period
matching	Text	No	Enum: intersects , encloses . The parameter defines a way how the start/end date of a statement is treated in comparison with From/To dates supplied in URL query. When intersects then any statement which period is anyhow overlapping the matching From/To period is selected. When enclosed then a statement which period falls fully inside matching from/to period is selected. When not supplied in URL enclosed is preferred as Default value.
availability	Text	No	Availability specifies whether the account statement is immediately available `IMMEDIATE` or has been archived by the bank `ARCHIVED`. If the value is not filled in, the default value `ALL` is used.
size	Number	No	Paging. Number of entries per page
page	Number	No	Paging. Required page. + Default: 0

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter 1.5.1 Request headers.

Conditional use of common header parameters:

- **Authorization** parameter is always mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter 1.5.2 Response headers.

The content of GET response for calling, please see Chapter 3.5.3.2 RESPONSE MESSAGE ELEMENTS Available Account Statements

HTTP STATUS	ERROR CODE	PURPOSE
-------------	------------	---------

CODE		
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
401	UNAUTHORISED	Invalid/missing certificate = provider is not authenticated
404	ID_NOT_FOUND	Invalid or unknown payment account identifier (id)
400	INVALID_AVAILABILITY	[InvalidInputAvailability] – possible values: IMMEDIATE, ARCHIVED, ALL
400	INVALID_PERIOD_SPECIFIED	Mismatched, invalid or incomplete parameters supplied for (year,month,quarter,week)x(day)x(from,to,matching)
400	INVALID_CURSOR	[InvalidCursor] – invalid pagination cursor.

3.5.3.2 RESPONSE MESSAGE ELEMENTS Available Account Statements

LEVEL	MESSAGE ELEMENT	OCCURENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
+	accountStatements	[1..1]	AISP	±	List of account statements for selected year.
++	accountStatementId	[1..1]	AISP	Text	Unique identifier of the accountStatement
++	year	[0..1]	AISP	number	Year of account statement. Format/value YYYY
++	halfyear	[0..1]	AISP	number	Halfyear of account statement. Number (values 1, 2)
++	quarter	[0..1]	AISP	number	Quarter of account statement in case of quartely frequency. format Q (1-4, e.g. 3 for July-September span)
++	month	[0..1]	AISP	number	Month of the account statement in case of monthly frequency. format M (e.g. 2 for

					February)
++	week	[0..1]	AISP	number	Week number of account statement in case of weekly frequency. format w (e.g. 1-53 for all weeks within year)
++	day	[0..1]	AISP	number	Day number of the statement with daily frequency, denoted as d (1-366)
++	sequenceNumber	[1..1]	AISP	number	The account statement's sequence number. e.g. 1
++	period	[1..1]	AISP	text	The period (i.e. "frequency" term) of requested statement. See chapter Account statement period.
++	dateFrom	[1..1]	AISP	ISO date	The start-date of the statement period. Format YYYY-MM-DD.
++	dateTo	[1..1]	AISP	ISO date	The end-date of the statement period. Format YYYY-MM-DD.
++	formats	[1..n]	AISP	±	Types of account statement's availability and and their formats.
+++	availability	[1..1]	AISP	text	Type of account statement's availability: The account

					statement is immediately available (IMMEDIATE), has been archived by the bank (ARCHIVED) or is not available through the API (UNAVAILABLE).
+++	format	[1..1]	AISP	text	Available media type listing formats. The element describes the content-type values that can be requested. example PDF, XML, ABO

Example request for statement from May 2022

```
GET
https://api.bank.cz/v1/my/accounts/048885570000001020045/statements?
year=2022&month=5&availability= IMMEDIATE&page=1&size=100
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:33:01 GMT
User-Involved: true
API-key: 00000000-1212-0e0e-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456
```

Example request for all statements from May 1 2022 to June 1 2022

```
GET
https://api.bank.cz/v1/my/accounts/04888557000001020045/statements?
from=2022-05-01&to=2022-06-01&matching=intersects&page=1&size=50

Content-Type: application/json

X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4

Date: Wed, 6 Jan 2019 07:33:01 GMT

User-Involved: true

API-key: 00000000-1212-0e0e-a0a0-123456789abc

TPP-Name: Google China

Authorization: Bearer AbCdEf123456
```

Example response

```
{
  "accountStatements": [
    {
      "accountStatementId": "aca74ae2-e0d3-4094-b574-63b0dc237941",
      "year": 2021,
      "month": 7,
      "sequenceNumber": 7,
      "period": "MONTH",
      "dateFrom": "2021-07-01",
      "dateTo": "2021-07-31",
      "formats": [
        {
          "availability": "IMMEDIATE",
          "format": "PDF"
        },
        {
          "availability": "IMMEDIATE",
          "format": "JSON"
        }
      ],
    },
  ],
}
```



```
    },
    {
      "accountStatementId": "12374ae2-e0d3-4094-b574-63b0dc237941",
      "year": 2021,
      "month": 7,
      "sequenceNumber": 3,
      "period": "DAY",
      "dateFrom": "2021-07-03",
      "dateTo": "2021-07-03",
      "formats": [
        {
          "availability": "IMMEDIATE",
          "format": "JSON"
        },
        {
          "availability": "IMMEDIATE",
          "format": "PDF"
        }
      ],
    }
  ],
  "links": [
    {
      "rel": "nextPage",
      "value": "3e7e6eb50120afca6a619fdddebef108ad99c271"
    }
  ]
]
```

3.5.4 Get account statement download Url (GET /my/accounts/{id}/statement/download{?accountStatementId})

Get the account statement download url.

Resource characteristics

Mandatory:	yes
URI:	/my/accounts/{id}/statement/download{?accountStatementId,format,feature}
HTTP Method:	GET
Authorization:	request requires the authorization of user/client as part of API calling
Use certificate:	request requires the use of the qualified third-party certificate
Paging:	No
Sorting:	No
Filtering:	No
Input character set:	no restrictions in relation to chapter 3.1.1

3.5.4.1 REQUEST STRUCTURE Download account statement file

Query parameters of the request:

PARAMETER	TYPE	MANDATORY	PURPOSE
id	Text	No	API payment account identifier from the response to a query on the list of accounts.
accountStatementId	Number	No	Unique account statement identifier.
format	Text	No	Chosen format of the statement, e.g. PDF, XML
feature	Text	No	Standardized or Bank-specific features (documented on the bank service level) - subtype of ABO, embedded or standalone JSON, ... etc.

Parameters of the request header:

No additional parameters defined in addition to the parameters mentioned in the chapter 1.5.1 Request headers.

Conditional use of common header parameters:

- **Authorization** parameter is always mandatory for this resource

Parameters of the response header:

No additional parameters defined in addition to the parameters mentioned in the chapter 1.5.2 Reponse headers.

The content of GET response for calling, please see Chapter 3.5.4.2 RESPONSE MESSAGE ELEMENTS
Download account statement file

Error codes defined for the GET service, List of client account statements:

<i>HTTP STATUS CODE</i>	<i>ERROR CODE</i>	<i>PURPOSE</i>
401	UNAUTHORISED	Invalid/missing access token = user is not authenticated
401	UNAUTHORISED	Invalid/missing certificate = provider is not authenticated
404	ID_NOT_FOUND	Invalid or unknown payment account identifier (id) or accountStatementId
404	FORMAT_NOT_AVAILABLE	Invalid or missing format
429	TOO_MANY_REQUESTS	For example. in case of too frequent queries for ARCHIVED statements.

3.5.4.2 RESPONSE MESSAGE ELEMENTS Download account statement file

<i>LEVEL</i>	<i>MESSAGE ELEMENT</i>	<i>OCCURENCE</i>	<i>PAYMENT TYPE</i>	<i>FORMAT TYPE</i>	<i>PRESENTATION</i>
+	pollInterval	[0..1]	AISP	Number	An interval in milliseconds that determines how long it is possible to ask again if the statement is available.
+	dataDownload URL	[0..1]	AISP	Text	URL for download Statement content in case of non-paged JSON

Example response in case of polling

```
{  
  "pollInterval": 5000  
}
```

Example response in case of polling end

```
{  
  "pollInterval": 5000,  
  "dataDownloadURL": "https://...."  
}
```

3.5.4.3 Response structure for JSON version of downloaded statement (response to obtained dataDownloadUrl).

LEVEL	MESSAGE ELEMENT	OCCURENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
+	exportStatement	[1..1]	AISP	±	Data structure for the format application/json.
++	accountPrefix	[0..1]	AISP	Text	BBAN account prefix
++	accountNumber	[1..1]	AISP	Text	BBAN account number
++	bankCode	[0..1]	AISP	Text	Code of the bank according to CNB's [ČÍSELNÍK KÓDŮ PLATEBNÍHO STYKU V ČESKÉ REPUBLICE (ČKPS)]
++	accountCurrency	[1..1]	AISP	Code	Currency code according to ISO 4217
++	product18N	[0..1]	AISP	Text	Product name.
++	begBalance	[0..1]	AISP	Number	Value of account balance at the beginning of the statement period. The currency corresponds to the currency of the account.
++	finBalance	[1..1]	AISP	Number	Value of account balance at the end of the statement period. The currency corresponds to the currency of the

					account.
++	period	[1..1]	AISP	Text	The period (frequency) of requested statement. See chapter Account statement period.
++	rejTurnCount	[0..1]	AISP	Number	Count of rejected transactions.
++	generateDate	[1..1]	AISP	ISO date	The date of the statement generation in YYYY-MM-DD format.
++	sequenceNumber	[1..1]	AISP	Number	The account statement's sequence number.
++	creditSum	[0..1]	AISP	Number	The sum of values of all credit transactions.
++	debitSum	[1..1]	AISP	Number	The sum of values of all debit transactions.
+	transactions	[0..1]	AISP	±	Response message elements as defined in chapter 3.1.5.1 Exception: creditor's name is optional for Account statement response.

Account statement - occurrence of elements in JSON version

LEVEL	MESSAGE ELEMENT	OCCURRENCE
+	entryReference	[0..1]
+	amount	[1..1]
++	value	[1..1]
++	currency	[1..1]
+	creditDebitIndicator	[1..1]
+	reversalIndicator	[0..1]
+	status	[1..1]
+	bookingDate	[1..1]
++	date	[1..1]
+	valueDate	[1..1]

++	date	[1..1]
+	bankTransactionCode	[1..1]
++	proprietary	[1..1]
+++	code	[1..1]
+++	issuer	[1..1]
+	entryDetails	[1..1]
++	transactionDetails	[1..1]
+++	references	[0..1]
++++	paymentId	[0..1]
++++	messageIdentification	[0..1]
++++	accountServicerReference	[0..1]
++++	paymentInformationIdentification	[0..1]
++++	instructionIdentification	[0..1]
++++	endToEndIdentification	[0..1]
++++	chequeNumber	[0..1]
++++	clearingSystemReference	[0..1]
+++	amountDetails	[0..1]
++++	instructedAmount	[0..1]
+++++	amount	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
++++	transactionAmount	[0..0]
+++++	amount	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]
++++	counterValueAmount	[0..1]
+++++	amount	[1..1]
++++++	value	[1..1]
++++++	currency	[1..1]

+++++	currencyExchange	[0..1]
+++++	sourceCurrency	[1..1]
+++++	targetCurrency	[0..1]
+++++	exchangeRate	[1..1]
++++	proprietaryAmount	[0..0]
+++++	type	[0..0]
+++++	amount	[0..0]
+++++	value	[1..1]
+++++	currency	[1..1]
+++	relatedParties	[1..1]
++++	debtor	[0..1]
+++++	name	[0..1]
++++	debtorAccount	[1..1]
+++++	identification	[1..1]
+++++	iban	[1..1]..either or other identification
+++++	other	[1..1].. either or IBAN
+++++	identification	[1..1]
++++	creditor	[0..1]
+++++	name	[0..1]
++++	creditorAccount	[1..1]
+++++	identification	[0..1]
+++++	iban	[0..1].. either or other identification
+++++	other	[0..1].. either or IBAN
+++++	identification	[1..1]
+++	relatedAgents	[1..1]
++++	debtorAgent	[1..1]
+++++	financialInstitutionIdentification	[1..1]
+++++	bic	[1..1]...either or other

		identification
+++++	name	[0..1]
+++++	other	[1..1]...either or BIC
+++++	identification	[1..1]
++++	creditorAgent	[1..1]
+++++	financialInstitutionIdentification	[1..1]
+++++	bic	[1..1]...either or other identification
+++++	name	[0..1]
+++++	other	[1..1]...either or BIC
+++++	identification	[1..1]
+++	remittanceInformation	[0..1]
++++	unstructured	[0..1]
++++	structured	[0..1]
+++++	creditorReferenceInformation	[0..1]
+++++	reference	[0..1]
+++	additionalTransactionInformation	[0..1]

Example response

```
{
  "exportStatement": {
    "accountPrefix": "0",
    "accountNumber": "0012085339",
    "bankCode": "0800",
    "accountCurrency": "CZK",
    "productName": "Běžný účet",
    "begBalance": "100.00",
    "finBalance": "200.00",
    "period": "DAY",
    "rejTurnCount": "0",
    "generateDate": "2022-01-01",
    "sequenceNumber": "1",
```



```
    "creditSum": "100.00",
    "debitSum": "0.00"
  },
  "pageNumber": 0
  "pageCount": 1
  "pageSize": 50
  "nextPage": 1
  "transactions": [
    {
      "entryReference": "2000007598293086",
      "amount": {
        "value": "100.00",
        "currency": "CZK"
      },
      "creditDebitIndicator": "CRDT",
      "status": "BOOK",
      "bookingDate": {
        "date": "2022-01-01"
      },
      "valueDate": {
        "date": "2022-01-01"
      },
      "bankTransactionCode": {
        "proprietary": {
          "code": 10000107000,
          "issuer": "Czech Banking Association"
        }
      },
      "entryDetails": {
        "transactionDetails": {
          "references": {
            "accountServicerReference": "TPPBNXAR43H "
          },
          "amountDetails": {
```

```
"instructedAmount": {
  "amount": {
    "value": "100.00",
    "currency": "CZK"
  }
},
"counterValueAmount": {
  "amount": {
    "value": "100.00",
    "currency": "CZK"
  }
},
"relatedParties": {
  "debtor": {
    "name": "Mgr. Vratislav Vlček"
  },
  "debtorAccount": {
    "identification": {
      "iban": "CZ6208000000000223253339",
      "other": {
        "identification": "0223253339/0800"
      }
    }
  },
  "creditor": {
    "name": "Řeháková Hana"
  },
  "creditorAccount": {
    "identification": {
      "iban": "CZ2808000000000120853339",
      "other": {
        "identification": "0012085339/0800"
      }
    }
  }
}
```

```

    }
  },
  "relatedAgents": {
    "creditorAgent": {
      "financialInstitutionIdentification": {
        "bic": "GIBACZPX",
      }
    },
    "debtorAgent": {
      "financialInstitutionIdentification": {
        "bic": "GIBACZPX",
      }
    },
    "additionalTransactionInformation": "Příchozí úhrada"
  }
}
]
}

```

3.5.5 Code lists for account statements

3.5.5.1 Account Statement Period

PERIOD		PERIOD DESIGNATION
Denní	Daily	DAY
Týdenní	Weekly	WEEK
Měsíční	Monthly	MONTH
Kvartální	Quarterly	QUARTER
Půlroční	Halfyear	HALF
Roční	Annualy	YEAR

3.5.5.2 Supported Account statement formats

FORMAT	BUSINESS CONTENT TYPE	TECHNICAL CONTENT TYPE	FEATURE
PDF	Pdf according to Portable Document Format specification	application/pdf	
XML	COBS XML Statement format	application/xml	
ABO	ABO format. It can have Bank specific features	application/octet-stream	
GEMINI	Gemini format. It can have Bank specific features	application/octet-stream	
JSON	COBS json format	application/json	<ul style="list-style-type: none"> ▪ JSON\$EMBD: Embedded in API response, possibly paged ▪ JSON\$STND: Standalone - supplied on download URL

4 Description of Elements of Messages for Services Payment Initiation, Account Information and Balance Check

4.1 Amount

JSON record: amount

Occurrence ALL: [1..1]

Definition: An amount of money that is to be transferred between the payer and the payee before deducting fees. It may be stated as only the Instructed Amount or the Equivalent Amount.

Type: This element for TUZEM and SEPA payments consists of the following elements of AmountType3CZ:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
++	InstructedAmount	[1..1]	Amount	Amount and currency in instruction

Type: This element for EHP and NONEHP payments consists of the following AmountType3Choice elements, i.e. can be implemented as a CHOICE between the following:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
++	InstructedAmount	[1..1]		Amount and currency in instruction
++	EquivalentAmount	[0..0]		Equivalent amount and currency

JSON example of element:

```

"amount": {
  "instructedAmount": {
    "value": 10050.15,
    "currency": "CZK"
  }
}

```

4.1.1 *InstructedAmount (Amount and currency in the instruction)*

JSON record: amount.instructedAmount

Occurrence ALL: [1..1]

Definition: The amount to be transferred between the payer and the recipient before charges

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++	value	[1..1]	Amount	Amount in instruction
+++	currency	[1..1]	Currency code ISO4217	Currency in instruction

Data type: Amount

Format: CurrencyAndAmount

4.1.1.1 Value (Amount of the transfer)

JSON record: amount.instructedAmount.value

Occurrence ALL: <1..1>

Definition: The amount to be transferred between the payer and the recipient before charges

Typ formátu: Amount

TUZEM: decimal places: 2, max. 1000000000000.00, min. 0.01

SEPA: decimal places: 2, max. 999999999.99, min. 0.01

EHP decimal places: 2, max. 999999999999999.99, min. 0.01

NONEHP: decimal places: 2, max. 999999999999999.99, min. 0.01

4.1.1.2 Currency (Měna převodu)

JSON record: amount.instructedAmount.currency

Occurrence ALL: <1..1>

Definition: The currency to which the amount in the instruction applies

Format: [A-Z]{3,3} - CurrencyCode, ISO 4217

TUZEM: CZK, or in another currency for payments within the bank, and only for the currency that the bank supports the payer.

SEPA: only for currency code EUR

EHP: only the currency the payer supports

NONEHP: only the currency the payer supports

4.1.2 EquivalentAmount (Equivalent amount and currency)

JSON record: amount.equivalentAmount

Occurrence ALL: <0..0>

Definition: The amount expressed in the currency of the payer's account, which is the equivalent of the amount to be transferred between the payer and the payee, before deducting the charges, expressed in the currency of the transfer. The element is only used for FX transactions.

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++	value	[1..1]	Amount	Equivalent transaction amount
+++	margin	{1..0}	Amount	Transaction conversion margin
+++	currency	[1..1]	Currency code ISO4217	Currency equivalent to transaction amount

Data type: Amount

Format: CurrencyAndAmount

4.2 authenticationMethod (Authentication method)

JSON record: authenticationMethod

Occurrence: [0..1]

Definition: Verification of the card holder

Application: The field is used if a card was used to verify sufficient account funds. It described the method and data that would be used for the transaction to verify the card holder.

Format type: CodeSet

CODE	NAME	DEFINITION
NPIN	OnLinePIN	On-line PIN authentication (PersonalIdentification Number).
PPSG	PaperSignature	Handwritten paper signature.
PSWD	Password	Authentication by a password.
SCRT	SecureCertificate	Electronic commerce transaction secured with the X.509 certificate of a customer.
SCNL	SecuredChannel	Channel-encrypted transaction.
SNCT	SecureNoCertificate	Secure electronic transaction without cardholder certificate.
CPSG	SignatureCapture	Electronic signature capture (handwritten signature).
ADDB	BillingAddressVerification	Cardholder billing address verification.
BIOM	Biometry	Biometric authentication of the cardholder
CDHI	CardholderIdentificationData	Cardholder data provided for verification, for instance social security number, driver license number, passport number.
CRYP	CryptogramVerification	Verification of a cryptogram generated by a chip card or another device, for instance ARQC (Authorisation Request Cryptogram).
CSCV	CSCVerification	Verification of Card Security Code.
PSVE	PassiveAuthentication	Authentication based on statistical cardholder behaviour.
CSEC	SecureElectronicCommerce	Authentication performed during a secure electronic commerce transaction.
ADDS	ShippingAddressVerification	Cardholder shipping address verification.

TOKP	PaymentToken	Verification or authentication related to the use of a payment token, for instance the validation of the authorised use of a token.
-------------	--------------	---

JSON example of element:

```
"authenticationMethod": "NPIN",
```

4.3 bankTransactionCode (Code of bank transaction)

JSON record: bankTransactionCode

Occurrence: [1..1]

Definition: The code of bank transaction according to the code list of the Czech Banking Association assigned to a specific payment. Use: Each bank uses its own detailed code list to identify payments, which is, however, based on the 1st-3rd level of the code list of transactions according to the CBA standard for camt.053.



Transaction codes:

PAYMENTS	
Transaction code	Description
10000101000	Domestic Payment - Outgoing domestic payment
10000102000	Domestic Payment - Outgoing domestic payment - express
10000103000	Domestic Payment - Collection - Credit
10000104000	Domestic Payments - Invoice - Debit
10000105000	Domestic Payment System - Standing Order

10000106000	Domestic Payment System - SIPO
10000107000	Domestic Payment System - Incoming domestic payment
10000201000	Foreign Payments - Outbound Foreign Payment
10000202000	Foreign Payments - Incoming foreign payment
10000301000	Others - Others
10000401000	SEPA CT - Outgoing SEPA payment
10000402000	SEPA CT - Outgoing SEPA payment - express
10000403000	SEPA CT - Incoming SEPA payment
10000501000	SEPA DD - Incoming SEPA DD Core
10000502000	SEPA DD - Outgoing SEPA DD Core
10000503000	SEPA DD - Incoming SEPA DD B2B
10000504000	SEPA DD - Outgoing SEPA DD B2B

CASH OPERATIONS

Transaction code	Description
20000101000	Deposit - Deposit without conversion in local currency
20000102000	Deposit - Deposit without conversion in foreign currency
20000103000	Deposit - Deposit with conversion
20000104000	Deposit - Deposit on account in another bank
20000201000	Selection - Choose without conversion in local currency
20000202000	Selection - Choose without conversion in a foreign currency
20000203000	Select - Choose with Conversion
20000301000	Other - Cash Processing
20000302000	Other - Other services

CARD OPERATIONS

Transaction code	Description
30000101000	ATM - ATM withdrawals
30000102000	ATM - Choice of ATMs abroad
30000103000	ATM – Deposit

30000201000	POS - Card payment at the merchant
30000202000	POS - Cashback
30000203000	POS - Cash advance
30000301000	Other - Other services

CHARGES	
Transaction code	Description
40000101000	Domestic Payment - Outgoing Payment
40000102000	Domestic Payment - Incoming Payment
40000103000	Domestic Payment System - Standing Order
40000104000	Domestic Payment System - Other
40000201000	Foreign Payments - Outgoing Payment
40000202000	Foreign Payments - Incoming Payment
40000203000	Foreign Payments - Others
40000301000	SEPA - Outgoing SEPA payment
40000302000	SEPA - Incoming SEPA payment
40000303000	SEPA - SEPA DD B2C
40000304000	SEPA - SEPA DD B2B
40000401000	Cash operations - Deposit
40000402000	Cash Operations - Selection
40000403000	Cash operations - Other
40000501000	Card Operations - ATM
40000502000	Card Operations - POS
40000503000	Card Operations - Others
40000601000	Others – Loans
40000602000	Others – Checks
40000603000	Others - Trade finance
40000604000	Others - Treasury
40000605000	Other - Other services

R-TRANSACTIONS	
Transaction code	Description
50000201001	SEPA DD - SEPA DD B2C - Return
50000201002	SEPA DD - SEPA DD B2C - Refund
50000201003	SEPA DD - SEPA DD B2C - Reversal
50000201004	SEPA DD - SEPA DD B2C - Reject
50000201005	SEPA DD - SEPA DD B2C - Refusal
50000201006	SEPA DD - SEPA DD B2C - Revocation
50000201007	SEPA DD - SEPA DD B2C - Request for cancellation
50000202001	SEPA DD - SEPA DD B2B - Return
50000202002	SEPA DD - SEPA DD B2B - Refund
50000202003	SEPA DD - SEPA DD B2B - Reversal
50000202004	SEPA DD - SEPA DD B2B - Reject
50000202005	SEPA DD - SEPA DD B2B - Refusal
50000202006	SEPA DD - SEPA DD B2B - Revocation
50000202007	SEPA DD - SEPA DD B2B - Request for cancellation

OTHERS	
Transaction code	Description
90000101000	Loans
90000201000	Interest
90000301000	Checks
90000401000	Trade finance
90000501000	Treasury
90000601000	Bonuses
90000701000	Other services

JSON example of element:

```
"bankTransactionCode": {
```

```

"proprietary": {
  "code": 00004000010,
  "issuer": "CBA"
}

```

4.4 bookingDate (Date of processing)

JSON record: bookingDate

Occurrence: [1..1]

Definition: Date of payment processing/posting by the bank.

Application: Depends on the transaction type and the method how the bank presents data (and time) of payment processing/posting

Type: This message element consists of the following elements:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
++	date	[1..1]	ISODate in format YYYY-MM-DD where: YYYY = four-digit year MM = two-digit month (01=January, etc.) DD = two-digit day of month (01 through 31) hh = two digits of hour (00 through 23) (am/pm NOT allowed) mm = two digits of minute (00	Date of payment processing/posting by the bank

			through 59) ss = two digits of second (00 through 59) TZD = time zone designator (Z or +hh:mm or -hh:mm)	
--	--	--	--	--

JSON example of element:

```

"bookingDate": {
  "date": "2016-09-05T00:00:00+01:00"
}

```

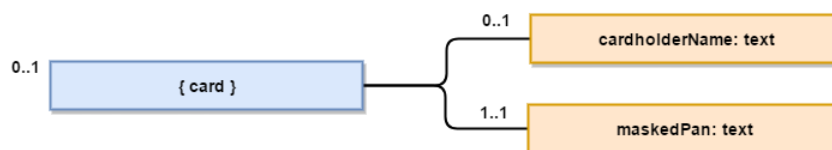
[ZÁKLADNÍ ELEMENTY ZPRÁVY](#)

4.5 card (Card)

JSON record: card

Occurrence: [0..1]

Definition: Payment card linked to the transaction, if the card was used for initiation



Type: This message element consists of the following elements:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
++	cardholderName	[0..1]	CISP	Max45Text	Card holder name
++	maskedPan	[1..1]	CISP	Max30Text	Masked card number

JSON example of element:

```
"card": {  
  "cardHolderName": "Jan Novák",  
  "maskedPAN": "1234*****6789"  
}
```

4.5.1 *cardholderName (Card holder name)*

JSON record: `card.cardholderName`

Occurrence: [0..1]

Definition: Card holder name

Application: The field contains the name of card holder stated on the card.

Format type: `Max45Text`

JSON example of element:

```
"cardHolderName": "Jan Novák"
```

4.5.2 *maskedPan (Masked card number)*

JSON record: `card.maskedPan`

Occurrence: [1..1]

Definition: Masked card number

Application: The field contains a partially masked card number, which is stated in the payment receipt or is displayed to the card holder after verifying the balance. Masked numbers may be replaced by spaces or the character „*“.

Format type: Max30Text

JSON example of element:

```
"maskedPAN": "1234*****6789"
```

4.6 code (the balance type in the account balance response)

JSON record: type.codeOrProprietary.code

Occurrence: [1..1]

Definition: Defines the type of stay shown in response when querying the balance on a specific account.

Application: Indicates the type of balance to which the balance information relates

Format type: Text

CODE	VALUE	DESCRIPTION
CLAV	ClosingAvailable	Available balance
PRCD	PreviouslyClosedBooked	Opening balance
CLBD	ClosingBooked	Closing balance
ITBD	InterimBooked	Intermediate balance

JSON example of element:

```
"type": {  
  "codeOrProprietary": {  
    "code": "PRCD"  
  }  
}
```

4.7 *creditDebitIndicator (Indication of debit/credit payment)*

JSON record: creditDebitIndicator

Occurrence: [1..1]

Definition: Indication of whether this is a debit payment or credit payment in the account.

Application: The indication of whether this is a debit or credit payment is marked with one of the following codes:

Format type: CreditDebitCode

CODE	DESCRIPTION
DBIT	To the debit of
CRDT	In other cases

JSON example of element:

```
"creditDebitIndicator": "CRDT"
```

4.8 *creditor (Payee)*

JSON record: creditor

Occurrence:

TUZEM (Domestic payments): [0..0]

SEPA (SEPA payments): [1..1]

EHP (Foreign payments within EEA): [1..1]

NONEHP (Foreign payments outside EEA): [1..1]

Definition: The party to which a financial amount is paid.

Type: This message element contains the following elements **PartyIdentification32CZ2**:

LEVEL	MESSAGE ELEMENT	FORMAT TYPE	PRESENTATION
++	Name	Max70Text	Payee name
++	Postal Address		Payee postal address

JSON example of element:

```
"creditor": {  
  "name": "1. wiena investment",  
  "postalAddress": {  
    "streetName": "Reisnerstrase",  
    "buildingNumber": "20",  
    "postCode": "1030",  
    "townName": "Wiena",  
    "country": "AT"  
  }  
}
```

4.8.1 name (Name)

JSON record: creditor.name

Occurrence:

TUZEM (Domestic payments): [0..0]

SEPA (SEPA payments): [1..1]

EHP (Foreign payments within EEA): [1..1]

NONEHP (Foreign payments outside EEA): [1..1]

Definition: Name under which the party is known, and which is normally used to identify the party.

Data type: Max70Text

JSON example of element:

```
"name": "1. wiena investment"
```

4.8.2 *postalAddress (Payee postal address)*

JSON record: *creditor.postalAddress*

Occurrence:

TUZEM (Domestic payments): [0..0]

SEPA (SEPA payments): [0..1]

EHP (Foreign payments within EEA): [0..1]

NONEHP (Foreign payments outside EEA): [1..1]

Definition: Information that localizes and identifies the specific address such as the postal address.

Type: This message element contains the following elements **PostalAddress6CZ**

JSON example of element:

```
"postalAddress": {  
  "streetName": "Reisnerstrase",  
  "buildingNumber": "20",  
  "postCode": "1030",  
  "townName": "Wiena",  
  "country": "AT"
```

```
}
```

4.8.2.1 streetName (Street)

JSON record: creditor.postalAddress.streetName

Occurrence:

TUZEM (Domestic payments): [0..0]

SEPA (SEPA payments): [0..1]

EHP (Foreign payments within EEA): [0..1]

NONEHP (Foreign payments outside EEA): [0..1]

Definition: Street name or section.

Data type: Max70Text

JSON example of element:

```
"streetName": "Reisnerstrase",
```

4.8.2.2 buildingNumber (Building number)

JSON record: creditor.postalAddress.buildingNumber

Occurrence:

TUZEM (Domestic payments): [0..0]

SEPA (SEPA payments): [0..1]

EHP (Foreign payments within EEA): [0..1]

NONEHP (Foreign payments outside EEA): [0..1]

Definition: Number that identifies the position of building in the street.

Data type: Max16Text

JSON example of element:

```
"buildingNumber": "20",
```

4.8.2.3 postCode (Postcode)

JSON record: creditor.postalAddress.postCode

Occurrence:

TUZEM (Domestic payments): [0..0]

SEPA (SEPA payments): [0..1]

EHP (Foreign payments within EEA): [0..1]

NONEHP (Foreign payments outside EEA): [0..1]

Definition: Identifier consisting of a group of letters and figures which are assigned to a postal address so that post may be sorted.

Data type: Max16Text

JSON example of element:

```
"postCode": "1030"
```

4.8.2.4 townName (Town)

JSON record: creditor.postalAddress.townName

Occurrence:

TUZEM (Domestic payments): [0..0]

SEPA (SEPA payments): [0..1]

EHP (Foreign payments within EEA): [0..1]

NONEHP (Foreign payments outside EEA): [0..1]

Definition: Name of the built-up area with defined boundaries and local self-administration.

Data type: Max35Text

JSON example of element:

```
"townName": "Wiena",
```

4.8.2.5 country (Country)

JSON record: creditor.postalAddress.country

Occurrence:

TUZEM (Domestic payments): [0..0]

SEPA (SEPA payments): [0..1]

EHP (Foreign payments within EEA): [0..1]

NONEHP (Foreign payments outside EEA): [0..1]

Definition: A country with its own government

Data type: CountryCode

Data format: [A-Z]{2,2}

Rule: Country code is checked against the list of country names under the ISO 3166 (2 alphanumeric characters code version).

JSON example of element:

```
"country": "AT"
```

4.8.2.6 addressLine (Unstructured address record)

JSON record: creditor.postalAddress.addressLine

Occurrence:

TUZEM (Domestic payments): [0..0]

SEPA (SEPA payments): [0..2]

EHP (Foreign payments within EEA): [0..2]

NONEHP (Foreign payments outside EEA): [0..2]

Definition: Information that localizes and identifies the specific address as defined by postal services, presented in the free text format.

Data type: Max70Text

Maximum 2 lines with 70 characters per line

4.9 creditorAccount (Payee account)

JSON record: creditorAccount

Occurrence All: [1..1]

Definition: Clear identification of the payee account to which the credit item will be transferred as the transaction result.

Type: This message element consists of the following elements **CashAccount16CZ**:

LEVEL	MESSAGE ELEMENT	FORMAT TYPE	PRESENTATION
++	identification	±	Identification
++	currency	±	Currency

JSON examples of element:

```
"creditorAccount": {
```

```

"identification": {
  "iban": "CZ3908000000000204533335"
},
"currency": "CZK"
},

```

```

"creditorAccount": {
  "identification": {
    "other": {
      "identification": "123/0800"
    }
  },
  "currency": "CZK"
},

```

4.9.1 *identification (Identification)*

JSON record: creditorAccount.identification

Occurrence All: [1..1]

Definition: A clear and unambiguous account identification between the account holder and the party administering the account.

Type: This message element consists of the following elements:

TUZEM (Domestic payments): **AccountIdentification4ChoiceCZ**

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
-------	-----------------	------------	-------------	--------------

+++	IBAN	[1..1]	IBAN2007Identifier	Account no. in the IBAN format
+++	other	[0..0]	±	Other format of account number

SEPA (SEPA payments): **AccountIdentification4CZ**

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++	IBAN	[1..1]	IBAN2007Identifier	Account no. in the IBAN format

EHP (Foreign payments within EEA): **AccountIdentification4ChoiceCZ**

NONEHP (Foreign payments outside EEA): **AccountIdentification4ChoiceCZ**

LEVEL	OR	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++	{Or	IBAN	[1..1]	IBAN2007Identifier	Account no. in the IBAN format
+++	Or}	Other	[1..1]	±	Other format of account number

JSON examples of element:

```
"identification": {
  "iban": "CZ3908000000000204533335"
}
```

JSON example of element:

```
"identification": {
  "other": {
    "identification": "204533335"
  }
}
```


4.9.1.1 IBAN (Account number in the IBAN format)

JSON record: creditorAccount.identification.iban

Occurrence All: [1..1]

Definition: The international account number format used by financial institutions for clear and unambiguous client account identification. A more detailed specification of the format and content IBAN is available in the standard ISO 13616.

Data type:

IBAN2007Identifier Format: [A-Z]{2,2}[0-9]{2,2}[a-zA-Z0-9]{1,30}, valid IBAN consists of all the following components: country code, check digit and account no. in local BBAN format.

Account no. in IBAN format, which is issued by the Czech bank has exactly 24 alphanumeric characters.

JSON example of element:

```
"iban": "CZ3908000000000204533335"
```

4.9.1.2 other (Other account number format)

JSON record: creditorAccount.identification.other

Occurrence only EHP (Foreign payments within EEA) and NONEHP (Foreign payments outside EEA): [1. .1]

Definition: Unique account identification assigned by the party administering the account, using the identification scheme.

Type: This message element consists of the following elements **GenericAccountIdentification1CZ:**

LEVEL	MESSAGE ELEMENT	FORMAT TYPE	PRESENTATION
++++	identification	Max34Text	Account no. in local format

JSON example of element:

```
"other": {  
  "identification": "123"  
}
```

4.9.1.2.1 identification (Account number in local BBAN format)

JSON record: creditorAccount.identification.other.identification

Occurrence only EHP (Foreign payments within EEA) and NONEHP (Foreign payments outside EEA): [1..1]

Definition: An account number used by financial institutions for clear and unambiguous identification of the client account. The account number is issued by the client's bank. The account number in the local BBAN format which is issued by the Czech bank has maximum 16 numeric characters, minimum 2 numeric characters. It consists of an antenumber, which has minimum 2 numeric characters, maximum 6 numeric characters, and of a basic form of the account no. which has minimum 2 numeric characters, maximum 10 numeric characters.

Data type: Max34Text

JSON example of element:

```
"identification": "123"
```

4.9.2 currency (Currency)

JSON record: creditorAccount.currency

Occurrence only TUZEM (Domestic payments): [0..1]

Definition: Identification of the currency in which the payer account is administered. If the currency is stated, it must correspond to the administration, otherwise it is rejected.

Data type: CurrencyCode according to ISO 4217

Format: [A-Z]{3,3}

JSON example of element:

```
"currency": "CZK"
```

4.10 creditorAgent (Payee bank)

JSON record: creditorAgent

Occurrence:

TUZEM (Domestic payments): [0..0]

SEPA (SEPA payments): [0..1]

EHP (Foreign payments within EEA): [1..1]

NONEHP (Foreign payments outside EEA): [1..1]

Definition: Financial institution that administers the payee account.

Type: This message element consists of the following elements

BranchAndFinancialInstitutionIdentification4CZ:

LEVEL	MESSAGE ELEMENT	FORMAT TYPE	PRESENTATION
++	Financial Institution Identification	±	Financial Institution Identification

JSON example of element:

```
"creditorAgent": {  
  "financialInstitutionIdentification": {  
    "bic": "ABNYUS33",  
    "name": "New York Commercial Bank",  
    "postalAddress": {  
      "streetName": "Merrick Avenue615",  
      "buildingNumber": "61511590-6644",
```

```
        "postCode": "11590-6644",
        "townName": "WESTBURY, NY",
        "country": "US"
    }
}
}
```

4.10.1 *financialInstitutionIdentification (Financial institution identification)*

JSON record: creditorAgent.financialInstitutionIdentification

Occurrence:

TUZEM (Domestic payments): [0..0]

SEPA (SEPA payments): [0..1]

EHP (Foreign payments within EEA): [1..1]

NONEHP (Foreign payments outside EEA): [1..1]

Definition: A unique and unambiguous financial institution identification assigned by the international standard.

Type: This message element consists of the following elements **FinancialInstitutionIdentification7CZ**

JSON example of element:

```
"financialInstitutionIdentification": {
  "bic": "ABNYUS33",
  "clearingSystemMemberIdentification": {
    "memberIdentification": "2700"
  },
  "name": "New York Commercial Bank",
  "postalAddress": {
```

```
    "streetName": "Merrick Avenue 615",
    "buildingNumber": "61511590-6644",
    "postCode": "11590-6644",
    "townName": "WESTBURY, NY",
    "country": "US"
  }
}
```

4.10.1.1 BIC (BIC / SWIFT bank code)

JSON record: creditorAgent.financialInstitutionIdentification.bic

Occurrence:

TUZEM (Domestic payments): [0..0]

SEPA (SEPA payments): [1..1]

EHP (Foreign payments within EEA): [1..1]

NONEHP (Foreign payments outside EEA): [0..1]

Definition: Bank Identifier Code. A code assigned to financial institutions by the Registration authority according to the international identification scheme as described in the latest version of the ISO 9362. The bank code in the format of BIC / SWIFT code has exactly 8 or exactly 11 alphanumeric characters.

Data type: BICIdentifier

JSON example of element:

```
"bic": "ABNYUS33"
```

4.10.1.2 clearingSystemMemberIdentification (Identification of Clearing system participant)

JSON record: creditorAgent.financialInstitutionIdentification.clearingSystemMemberIdentification

Occurrence: only NONEHP (Foreign payments outside EEA): [0..1]

Definition: Identification of participant in the local clearing system

Type: This message element consists of the following elements **ClearingSystemMemberIdentification2**

JSON example of element:

```
"clearingSystemMemberIdentification": {  
  "memberIdentification": "2700"  
}
```

4.10.1.2.1 *clearingSystemIdentification (Clearing system identification)*

JSON record:

creditorAgent.financialInstitutionIdentification.clearingSystemMemberIdentification.clearingSystemIdentification

Occurrence: only NONEHP (Foreign payments outside EEA): [0..1]

Definition: Clearing system identification

Type: This message element consists of the following elements **ClearingSystemIdentification2Choice**

4.10.1.2.1.1 code (Code)

JSON record:

creditorAgent.financialInstitutionIdentification.clearingSystemMemberIdentification.clearingSystemIdentification.code

Occurrence: only NONEHP (Foreign payments outside EEA): [1..1] / either code or proprietary

Definition: A code identifying the local clearing system as stated in the external code list.

Data type: ExternalClearingSystemIdentification1Code

4.10.1.2.1.2 proprietary (Free format)

JSON record:

creditorAgent.financialInstitutionIdentification.clearingSystemMemberIdentification.clearingSystemIdentification.proprietary

Occurrence: only NONEHP (Foreign payments outside EEA): [1..1] / either code or proprietary

Definition: Identification of the local clearing system, in free format.

Data type: Max35Text

4.10.1.2.2 memberIdentification (Participant's identification code – clearing code)

JSON record:

creditorAgent.financialInstitutionIdentification.clearingSystemMemberIdentification.memberIdentification

Occurrence: only NONEHP (Foreign payments outside EEA): [1..1]

Definition: Participant's identification code or bank's clearing code.

Data type: Max35Text

JSON example of element:

```
"memberIdentification": "2700"
```

4.10.1.3 name (Name)

JSON record: creditorAgent.financialInstitutionIdentification.name

Occurrence: only NONEHP (Foreign payments outside EEA): [0..1]

Definition: Name under which the party is known, and which is normally used to identify that party.

Data type: Max70Text

JSON example of element:

```
"name": "New York Commercial Bank",
```

4.10.1.4 postalAddress (Postal address)

JSON record: creditorAgent.financialInstitutionIdentification.postalAddress

Occurrence: only NONEHP (Foreign payments outside EEA): [0..1]

Definition: Information that localizes and identifies the specific address as a postal address.

Type: This message element consists of the following elements **PostalAddress6CZ**

For foreign payments, it is recommended to fill in the unstructured address form. It is recommended to use / fill in the Country field from the Country field and up to two address lines from the Address Line field. Typically, the first line shows the street and the Land registry no., the city and city's postal code in the second line.

JSON example of element:

```
"postalAddress": {  
  "streetName": "Merrick Avenue 615",  
  "buildingNumber": "61511590-6644",  
  "postCode": "11590-6644",  
  "townName": "WESTBURY, NY",  
  "country": "US"  
}
```

4.10.1.4.1 streetName (Street)

JSON record: creditorAgent.financialInstitutionIdentification.postalAddress.streetName

Occurrence only NONEHP (Foreign payments outside EEA): [0..1]

Definition: Street name or section.

Data type: Max70Text

JSON example of element:

```
"streetName": "Merrick Avenue 615"
```

4.10.1.4.2 buildingNumber (Building no.)

JSON record: creditorAgent.financialInstitutionIdentification.postalAddress.buildingNumber

Occurrence only NONEHP (Foreign payments outside EEA): [0..1]

Definition: Number that identifies the position of a building in the street.

Data type: Max16Text

JSON example of element:

```
"buildingNumber": "61511590-6644"
```

4.10.1.4.3 postCode (Postcode)

JSON record: creditorAgent.financialInstitutionIdentification.postalAddress.postCode

Occurrence only NONEHP (Foreign payments outside EEA): [0..1]

Definition: Identifier consisting of a group of letters and figures which are assigned to a postal address so that post may be sorted.

Data type: Max16Text

JSON example of element:

```
"postCode": "11590-6644"
```

4.10.1.4.4 *townName (Town)*

JSON record: creditorAgent.financialInstitutionIdentification.postalAddress.townName

Occurrence only NONEHP (Foreign payments outside EEA): [0..1]

Definition: Name of the built-up area with defined boundaries and local self-administration.

Data type: Max35Text

JSON example of element:

```
"townName": "WESTBURY, NY"
```

4.10.1.4.5 *country (Country)*

JSON record: creditorAgent.financialInstitutionIdentification.postalAddress.country

Occurrence only NONEHP (Foreign payments outside EEA): [0..1]

Definition: A country with its own government

Data type: CountryCode corresponding to ISO 3166 (2 alphanumeric characters code version)

JSON example of element:

```
"country": "US"
```

4.10.1.4.6 *addressLine (Unstructured address record)*

JSON record: creditorAgent.financialInstitutionIdentification.postalAddress.addressLine

Occurrence only NONEHP (Foreign payments outside EEA): [0..2]

Definition: Information that localizes and identifies the specific address as defined by postal services, presented in the free text format.

Data type: Max70Text

Maximum 2 lines with 70 characters per line

4.10.1.5 other (Another bank identification)

JSON record: creditorAgent.financialInstitutionIdentification.other

Occurrence only NONEHP (Foreign payments outside EEA): [0..1]

Definition: Unique agent identification assigned to the institution using the identification scheme

Type: This message element consists of the following elements **GenericFinancialIdentification1CZ**

JSON example of element:

```
"other": {  
  "identification": "123"  
}
```

4.10.1.5.1 identification (Local bank code)

JSON record: creditorAgent.financialInstitutionIdentification.other.identification

Occurrence only NONEHP (Foreign payments outside EEA): [0..1]

Definition: Local format of the bank code

Data type: Max35Text

JSON example of element:

```
"identification": "123"
```

4.11 debtor (Payer)

JSON record: debtor

Occurrence All: [0..0]

Definition: A party that owes money to the payee.

Type: This message element consists of the following elements **PartyIdentification32CZ2**:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
++	Name	[0..0]	Max70Text	Payer name
++	Postal Address	[0..0]	±	Payer postal address

JSON example of element:

```
"debtor": {  
  "name": "RENWORTH s.r.o",  
  "postalAddress": {  
    "streetName": "Merrick Avenue 615",  
    "buildingNumber": "61511590-6644",  
    "postCode": "11590-6644",  
    "townName": "WESTBURY, NY",  
    "country": "US"  
  }  
},
```

4.11.1 name (Name)

JSON record: debtor.name

Occurrence All: [0..0]

Definition: Name under which the party is known, and which is normally used to identify that party.

Data type: Max70Text

Payer name is not normally filled in, unless agreed otherwise with the payer bank.

JSON example of element:

```
"name": "RENWORTH s.r.o"
```

4.11.2 *postalAddress (Postal address of the original payer)*

JSON record: debtor.postalAddress

Occurrence All: [0..0]

Definition: Information that localizes and identifies the specific address as a postal address.

Type: This message element consists of the following elements **PostalAddress6CZ**.

The payer address is not normally filled in, unless otherwise agreed with the payer bank. In this case, you can fill it in for domestic payments in a structured or unstructured format. If you use unstructured format, it is recommended to fill in two repeated Address Lines. Typically, the first line shows the street and the Land registry no., and the second line the town and the postal code. For SEPA payments and foreign payments, it is recommended to fill in the unstructured address form. It is recommended to use / fill in the Country field from the Country field and up to two lines of address from the Address Line field. Typically, the first line shows the street and the Land registry no., and the second line the town and the postal code.

JSON example of element:

```
"postalAddress": {  
  "streetName": "Merrick Avenue 615",  
  "buildingNumber": "61511590-6644",  
  "postCode": "11590-6644",  
  "townName": "WESTBURY, NY",
```

```
"country": "US"  
}
```

4.11.2.1 streetName (Street)

JSON record: debtor.postalAddress.streetName

Occurrence All: [0..0]

Definition: Street name or section.

Data type: Max70Text

JSON example of element:

```
"streetName": "Merrick Avenue 615"
```

4.11.2.2 buildingNumber (Building number)

JSON record: debtor.postalAddress.buildingNumber

Occurrence All: [0..0]

Definition: Number that identifies the position of building in the street.

Data type: Max16Text

JSON example of element:

```
"buildingNumber": "61511590-6644"
```

4.11.2.3 postCode (Postcode)

JSON record: debtor.postalAddress.postCode

Occurrence All: [0..0]

Definition: Identifier consisting of a group of letters and figures which are assigned to a postal address so that post may be sorted.

Data type: Max16Text

JSON example of element:

```
"postCode": "11590-6644"
```

4.11.2.4 townName (Town)

JSON record: debtor.postalAddress.townName

Occurrence All: [0..0]

Definition: Name of the built-up area with defined boundaries and local self-administration.

Data type: Max35Text

JSON example of element:

```
"townName": "WESTBURY, NY"
```

4.11.2.5 country (Country)

JSON record: debtor.postalAddress.country

Occurrence All: [0..0]

Definition: A country with its own government

Data type: CountryCode

Data format: [A-Z]{2,2}

Rule: Country code is checked against the list of country names under the ISO 3166 (2 alphanumeric characters code version).

JSON example of element:

```
"country": "US"
```

4.11.2.6 addressLine (Unstructured address record)

JSON record: debtor.postalAddress.addressLine

Occurrence All: [0..0]

Definition: Information that localizes and identifies the specific address as defined by postal services, presented in the free text format.

Data type: Max70Text

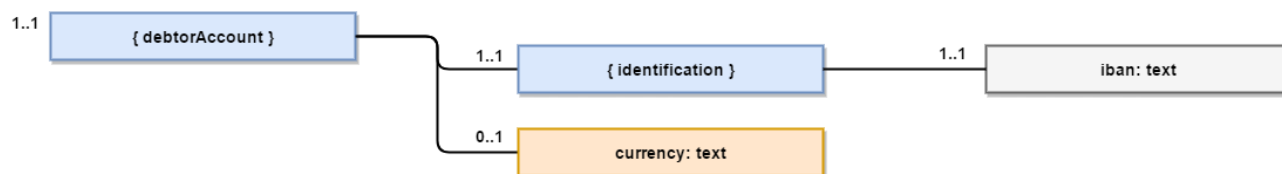
Maximum 2 lines with 70 characters per line

4.12 debtorAccount (Payer account)

JSON record: debtorAccount

Occurrence: [1..1]

Definition: Clear identification of the payer account in which a debit item will be executed as the transaction result.



Type: This message element consists of the following elements **CashAccount16CZ**:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
++	Chyba! Nenalezen z droj odkazů.	[1..1]	CISP	±	Payer account identification
++	currency	[0..1]	CISP	CurrencyCode, ISO 4217	Payer account currency

JSON example of element:

```

"debtorAccount": {
  "identification": {
    "iban": "CZ0708000000001019382023"
  },
  "currency": "CZK"
}

```

4.12.1 identification (Identification)

JSON record: *debtorAccount.identification*

Occurrence: [1..1]

Definition: Identification of payer account in which the Balance Check are verified. Only one type of identification is possible. Other formats are not permitted.

Type: This message element consists of the following elements **AccountIdentification4ChoiceCZ**:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++	IBAN	[1..1]	IBAN2007IdentifierCZ	Account no. in the IBAN format

+++	Other	[0..0]	±	Other format of account number
-----	-------	--------	---	--------------------------------

JSON examples of element:

```
"identification": {
  "iban": "CZ0827000000002108589434"
}
```

```
"identification": {
  "other": {
    "identification": "0000002108589434"
  }
}
```

4.12.1.1 IBAN (Account number in the IBAN format)

JSON record: debtorAccount.identification.iban

Occurrence: [1..1]

Definition: Payer account number in which Balance Check are verified, in the IBAN format

Format type: ISO 13616

Has the following structure:

- 2 characters (position 1-2) – country code „CZ“
- 2 characters (position 3-4) – check digit for module 97 – allows programme check of the number – protection against wrong entered account no. (e.g., resulting from any typing error)
- 4 characters (position 5-8) – payment system code – code assigned to the bank in the „Code list of payment system in the Czech Republic maintained by the CNB“

- 16 characters (position 9-24) – account no. in the format according to the Decree No. 169/2011 Coll.
- Maximum length 34 characters

[A-Z]{2,2}[0-9]{2,2}[a-zA-Z0-9]{1,30}, valid IBAN consists of all the following components: country code, check digit and account no. in local BBAN format.

Other formats for account identification are not permitted.

Data type: IBAN2007Identifier

JSON example of element:

```
"iban": "CZ0708000000001019382023"
```

4.12.1.2 other (Other account number format)

JSON record: debtorAccount.identification.other

Occurrence All: [0..0]

Definition: Unique account identification assigned by the party administering the account, using the identification scheme.

Type: This message element consists of the following elements **GenericAccountIdentification1CZ:**

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++	Identification	[0..0]	Max34Text	Account no. in local format

JSON example of element:

```
"other": {
  "identification": "0000002108589434"
}
```

4.12.1.2.1 *identification (Account number in local BBAN format)*

JSON record: debtorAccount.identification.other.identification

Occurrence All: [0..0]

Definition: An account number used by financial institutions for clear and unambiguous identification of the client account. The account number is issued by the client's bank. The account number in the local BBAN format which is issued by the Czech bank has maximum 16 numeric characters, minimum 2 numeric characters. It consists of an antenumber, which has minimum 2 numeric characters, maximum 6 numeric characters, and of a basic form of the account no. which has minimum 2 numeric characters, maximum 10 numeric characters.

Data type: Max16Num

JSON example of element:

```
"identification": "0000002108589434"
```

4.12.2 *currency (Currency)*

JSON record: debtorAccount.currency

Occurrence: [0..1]

Definition: Currency of the payer account in which Balance Check are verified

Application: The field defines the account currency in case that the debit account is multi-currency.

Format type: CurrencyCode, code under the ISO-4217, [A-Z]{3,3}

JSON example of element:

```
"currency": "CZK"
```

4.13 SEPA identification (Identification)

For SEPA payments, the payer, the payee, the original payer and the final payee the client has the option to pass the identification based on whether it is the organization identification or identification of a natural person.

The payee demands this information as the identification of the (original) payer and the identification of the (final) payee:

- Identification type – Organization / Natural person
- Detailed information
 - Either BIC / SWIFT code
 - Or Other identification with these data:
 - Document type
 - Document issuer

JSON record: identification

Occurrence only SEPA payments: [0..1]

Definition: A unique and unambiguous identification of the party.

Type: This message element consists of the following elements **Party6Choice**:

LEVEL	OR	MESSAGE ELEMENT	OCCURRENCE	PRESENTATION
+++	{Or	Organisation Identification	[1..1]	Organization identification
+++	Or}	Private Identification	[1..1]	Private individual identification

JSON example of element privateIdentification:

```
"identification": {  
  "privateIdentification": {  
    "other": [  
      {  
        "identification": "57842578842",
```

```

    "schemeName": {
      "proprietary": "passport",
      "issuer": "WIENA"
    }
  }
]
}
}

```

JSON examples of element organisationIdentification:

```

"identification": {
  "organisationIdentification": {
    "other": [
      {
        "identification": "48135283",
        "schemeName": {
          "proprietary": "partnership agreement"
        }
      }
    ]
  }
}

```

4.13.1 organisationIdentification (Organization identification)

JSON record: identification.organisationIdentification

Occurrence only SEPA payments: [1..1]

Definition: A unique and irreplaceable method for identifying an organization.

Type: This message element consists of the following elements **OrganisationIdentification4CZ:**

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
++++	BIC Or BEI	[0..1]	BICIdentifier	BIC / SWIFT code
++++	Other	[0..1]	±	Other identification

JSON example of element:

```
"organisationIdentification": {  
  "other": [  
    {  
      "identification": "48135283",  
      "schemeName": {  
        "proprietary": "partnership agreement"  
      }  
    }  
  ]  
}
```

4.13.1.1 bicOrBei (BIC / SWIFT code)

JSON record: identification.organisationIdentification.bicOrBei

Occurrence only SEPA payments: [0..1]

Definition: Bank Identifier Code. A code assigned to financial institutions by the Registration authority according to the international identification scheme as described in the latest version of the ISO 9362. An 8 or 11-digit bank code in the format of BIC / SWIFT code.

Data type: BICIdentifier

Format: [A-Z]{6,6}[A-Z2-9][A-NP-Z0-9]([A-Z0-9]{3,3}){0,1}

Rule: Valid BICs are registered at the ISO 9362 Registration authority and consist of eight (8) or eleven (11) successive characters containing the first three or all four of the following components: bank code, country code and location code are mandatory, while the branch code is optional.

JSON example of element:

```
"bicOrBei": "ABNYUS33"
```

4.13.1.2 other (Other identification)

JSON record: identification.organisationIdentification.other

Occurrence only SEPA payments: [0..1]

Definition: A unique identification of the organization assigned by the institution using an identification scheme.

Type: This message element consists of the following elements **GenericOrganisationIdentification1:**

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++++	identification	[1..1]	Max35Text	Identification data
+++++	schemeName	[0..1]	±	Document type

JSON example of element:

```
"other": [  
  {  
    "identification": "48135283",  
    "schemeName": {  
      "proprietary": "partnership agreement"  
    }  
  }  
]
```


4.13.1.2.1 *identification (Identification data)*

JSON record: identification.organisationIdentification.other.identification

Occurrence only SEPA payments: [1..1]

Definition: Identification assigned by the institution issuing the document.

Data type: Max35Text

JSON example of element:

```
"identification": "48135283"
```

4.13.1.2.2 *schemeName (Document type)*

JSON record: identification.organisationIdentification.other.schemeName

Occurrence only SEPA payments: [0..1]

Definition: ID document name.

Type: This message element is a **CHOICE** between the following elements

OrganisationIdentificationSchemeName1CZ:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++++	code	[1..1]...either or proprietary	Code	Code type for the payer / payee identification as an organization / legal entity in the form of code according to the ISO code list.
+++++	proprietary	[1..1]...either or code	Max35Text	Document type for the payer / payee identification as an organization / legal entity in free text format.

JSON example of element:

```
"schemeName": {  
  "proprietary": "partnership agreement"  
}
```

4.13.1.2.2.1 code (Code format)

JSON record: identification.privateIdentification.other.schemeName.code

Occurrence only SEPA payments: [1..1], **choice** between this element and proprietary

Definition: Name of the document type, in the form of a code according to the ISO code list.

Data type: Code

JSON example of element:

```
"code": "1.2.203.48135283"
```

4.13.1.2.2.2 proprietary (Free format)

JSON record: identification.organisationIdentification.other.schemeName.proprietary

Occurrence only SEPA payments: [1..1], **choice** between this element and code

Definition: Name of document type, in the free text format.

Data type: Max35Text

JSON example of element:

```
"proprietary": "RENWORTH s.r.o"
```

4.13.1.2.2.3 issuer (Document issuer)

JSON record: identification.organisationIdentification.other.schemeName.issuer

Occurrence only SEPA payments: [0..1]

Definition: Name of document issuer, i.e. the entity assigning the identification.

Data type: Max35Text

JSON example of element:

```
"issuer": "WIENA"
```

4.13.2 privateIdentification (Private individual identification)

JSON record: identification.privateIdentification

Occurrence only SEPA payments: [1..1]

Definition: A unique and irreplaceable method for identifying a natural person.

Type: This message element consists of the following elements **PersonIdentification5CZ**:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
++++	Other	[0..1]	Max35Text	Other identification

JSON example of element:

```
"identification": {  
  "privateIdentification": {  
    "other": [  
      {  
        "identification": "57842578842",  
        "schemeName": {  
          "proprietary": "passport",
```

```

        "issuer": "WIENA"
      }
    }
  ]
}

```

4.13.2.1 other (Other identification)

JSON record: identification.privateIdentification.other

Occurrence only SEPA payments: [0..1]

Definition: A unique identification of the person assigned by the institution using an identification scheme.

Type: This message element consists of the following elements **GenericPersonIdentification1**

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++++	identification	[1..1]	Max35Text	Identification data
+++++	schemeName	[0..1]	±	Document type

JSON example of element:

```

"other": [
  {
    "identification": "57842578842",
    "schemeName": {
      "proprietary": "passport",
      "issuer": "WIENA"
    }
  }
]

```

4.13.2.1.1 *identification (Identification data)*

JSON record: identification.privatIdentification.other.identification

Occurrence only SEPA payments: [1..1]

Definition: A unique and irreplaceable method for identifying a person.

Data type: Max35Text

JSON example of element:

```
"identification": "12356879131",
```

4.13.2.1.2 *schemeName (Document type)*

JSON record: identification.privatIdentification.other.schemeName

Occurrence only SEPA payments: [0..1]

Definition: Name of the identification document.

Type: This message element is a **CHOICE** between the following elements

PersonIdentificationSchemeName1Choice:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++++	code	[1..1]...either or proprietary	Code	Code type for the payer / payee identification as a physical person in the form of a code according to the ISO code list.
+++++	proprietary	[1..1]...either or code	Max35Text	Code type for the payer / payee identification as a physical person in free text format.

JSON example of element:

```
"schemeName": {  
  "proprietary": "passport",  
  "issuer": "WIENA"  
}
```

4.13.2.1.2.1 code (Code format)

JSON record: identification.privateIdentification.other.schemeName.code

Occurrence only SEPA payments: [1..1], **choice** between this element and proprietary

Definition: Name of the document type, in the form of a code according to the ISO code list.

Data type: Code

JSON example of element:

```
"code": "1.2.203.48135283"
```

4.13.2.1.2.2 proprietary (Free format)

JSON record: identification.privateIdentification.other.schemeName.proprietary

Occurrence only SEPA payments: [1..1], **choice** between this element and code

Definition: Name of the document type, in the free text format.

Data type: Max35Text

JSON example of element:

```
"proprietary": "passport"
```

4.13.2.1.2.3 issuer (Document issuer)

JSON record: identification.privateIdentification.other.schemeName.issuer

Occurrence only SEPA payments: [0..1]

Definition: Name of the document issuer, i.e. the entity assigning the identification.

Data type: Max35Text

JSON example of element:

```
"issuer": "WIENA"
```

4.14 entryDetails (Entry details)

JSON record: entryDetails

Occurrence: [0..n]

Definition: Entry details.

Application: This level only repeats once for the relevant item



Type: This message element consists of the following elements:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
++	transactionDetails	[0..n]	±	Payment detail

JSON example of element:

```
"entryDetails": {  
  "transactionDetails": {
```

```
    "amountDetails": {
      "instructedAmount": {
        "amount": {
          "value": 122.22,
          "currency": "CZK"
        }
      }
    },
    "additionalTransactionInformation": "PŘIPSÁNÍ ÚROKU ZE
ZUSTATKU"
  }
}
```

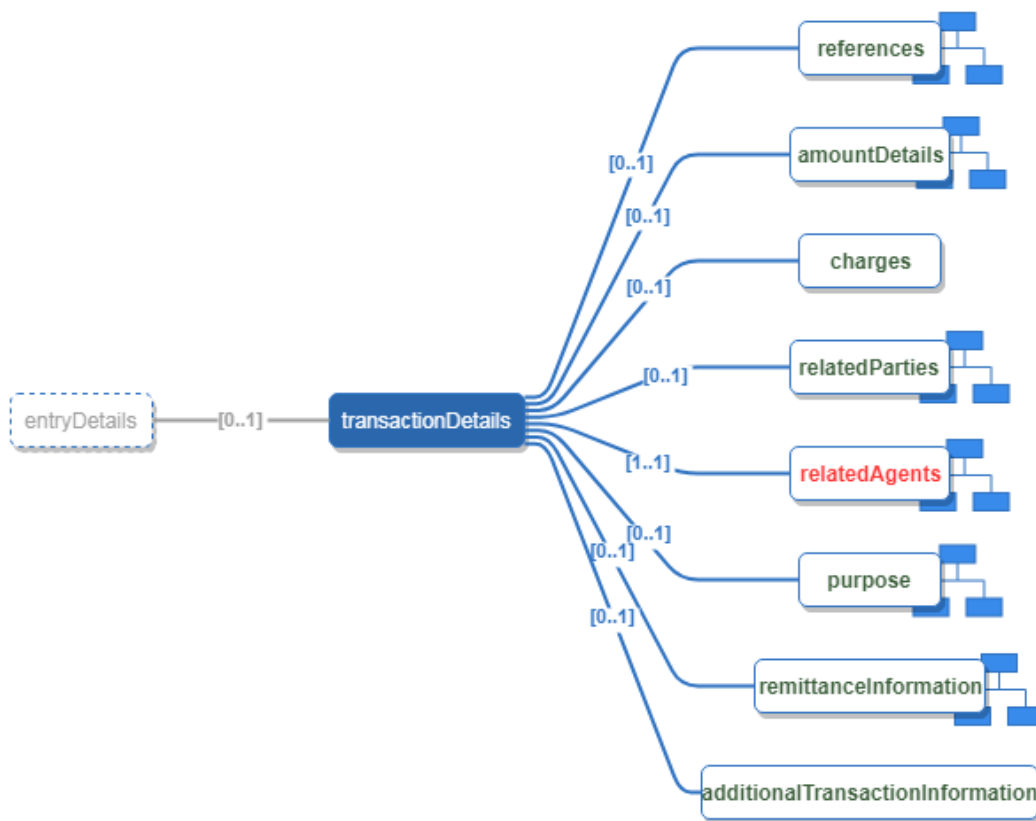
4.14.1 *transactionDetails (Payment details)*

JSON record: entryDetails.transactionDetails

Occurrence: [0..n]

Definition: Payment details.

Application: This level only repeats once for the relevant item.



Type: This message element consists of the following elements:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++	references	[0..1]	±	Set of references
+++	amountDetails	[0..1]	±	Payment details
+++	charges	[0..1]	±	Information on fees
+++	relatedParties	[0..1]	±	Information on payer
+++	relatedAgents	[1..1]	±	Information on payer bank
+++	purpose	[0..1]	±	
+++	remittanceInformation	[0..1]	±	
+++	additionalTransactionInformation	[0..1]	Max500Text	Additional information

JSON example of element:

```
"transactionDetails": {
  "amountDetails": {
```

```
    "instructedAmount": {
      "amount": {
        "value": 122.22,
        "currency": "CZK"
      }
    },
    "additionalTransactionInformation": "PŘIPSÁNÍ ÚROKU ZE ZUSTATKU"
  }
```

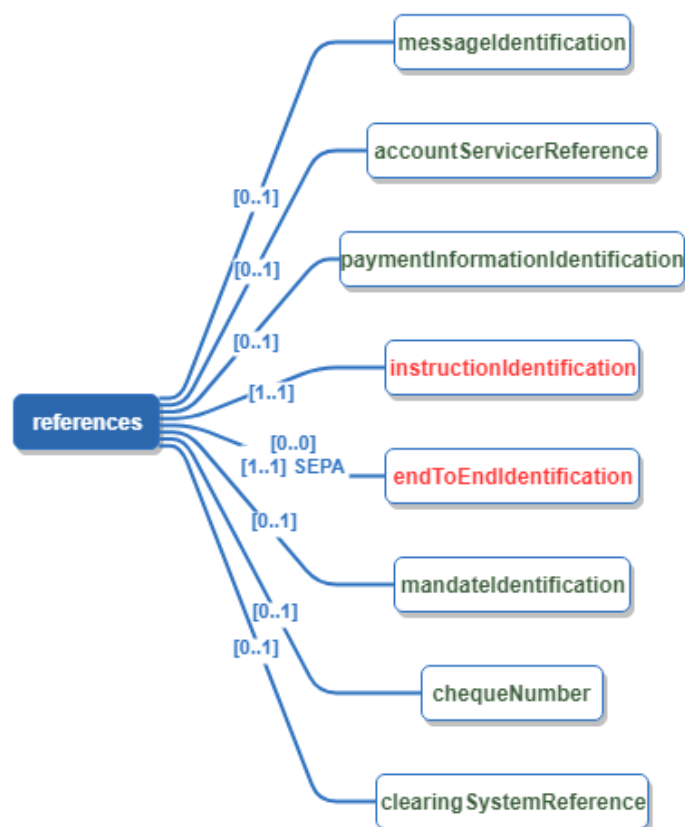
4.14.1.1 references (Reference)

JSON entry: entryDetails.transactionDetails.references

Occurrence: [0..1]

Definition: Set of references unambiguously identifying the payment.

Application: According to technical possibilities of the bank, it contains references of bank processing systems or client references.



Type: This message element consists of the following elements:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
++++	messageIdentification	[0..1]	Max35Text	Payment identification
++++	accountServicerReference	[0..1]	Max35Text	Bank payment reference
++++	paymentInformationIdentification	[0..1]	Max35Text	Payment identification
++++	instructionIdentification	[1..1]	Max35Text	Third-party payment identification
++++	endToEndIdentification	[0..0] [1..1] SEPA	Max35Text	Unique identification specified by the initiating client
++++	mandateIdentification	[0..0] [1..1]	Max35Text	Mandate reference
++++	chequeNumber	[0..1]	Max35Text	Cheque no.
++++	clearingSystemReference	[0..1]	Max35Text	Clearing reference

JSON example of element:

```
"references": {  
  "endToEndIdentification": "VS0250117002/SS0000000000/KS0000"  
}
```

4.14.1.1.1 *messageIdentification (Assumed payment identification)*

JSON record: entryDetails.transactionDetails.references.messageIdentification

Occurrence: [0..1]

Definition: Payment identification

Application: the assumed identification of a payment entered by a client on its initiation or order of the payment in the statement of payment history.

Format type: Max35Text

4.14.1.1.2 *accountServicerReference (Bank payment reference)*

JSON record: entryDetails.transactionDetails.references.accountServicerReference

Occurrence: [0..1]

Definition: Payment identification

Application: A bank reference assigned to the payment, e.g. on the initiation through direct banking.

Format type: Max35Text

4.14.1.1.3 *paymentInformationIdentification (Payment identification)*

JSON record: entryDetails.transactionDetails.references.paymentInformationIdentification

Occurrence: [0..1]

Definition: Payment identification

Application: A bank reference assigned to the payment by the bank. For card payments, the sequence no. of payment card may be added. For a domestic payment, the specific symbol may be entered.

Format type: Max35Text

4.14.1.1.4 *mandateIdentification (Mandate reference)*

JSON record: entryDetails.transactionDetails.references.mandateIdentification

Occurrence: [0..0], for SDD (Sepa Direct Debit) the occurrence is mandatory [1..1]

Definition: Identification of SDD mandate

Application: For payment processed in the SEPA Direct Debit scheme, a mandate reference is mandatory in the field, through which the client gave consent to debit the account.

Format type: Max35Text

4.14.1.1.5 *chequeNumber (Cheque number)*

JSON record: entryDetails.transactionDetails.references.chequeNumber

Occurrence: [0..1]

Definition: Payment identification

Application: For cheque transactions, the cheque no. may be stated here. For card transactions, a hidden card no. may be stated with asterisks, i.e. the card no. is given in the format xxxxxxxxxxxx1234.

Format type: Max35Text

4.14.1.1.6 *clearingSystemReference (Clearing reference)*

JSON record: entryDetails.transactionDetails.references.clearingSystemReference

Occurrence: [0..1]

Definition: Payment identification

Application: A code list value defined by the bank, identifying the payment type or the payment type name used. For card transactions, the card association may be stated.

Format type: Max35Text

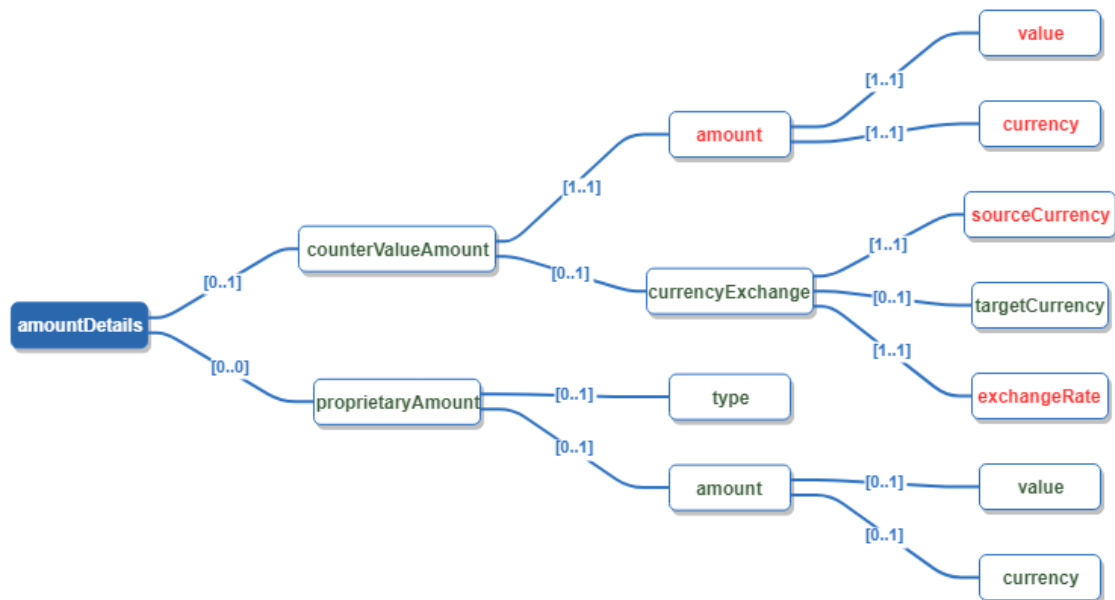
4.14.1.2 amountDetails (Amount details)

JSON record: entryDetails.transactionDetails.amountDetails

Occurrence: [0..1]

Definition: Payment amount specification

Application: Details for payment amount, mainly for conversion payment or cashback.



Type: This message element consists of the following elements:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
++++	instructedAmount	[0..1]	±	Amount of the requested payment to be transferred
++++	transactionAmount	[0..0]	±	Transaction amount
++++	counterValueAmount	[0..1]	±	Converted amount
++++	proprietaryAmount	[0..0]	±	Cashback amount

JSON example of element:

```

"amountDetails": {
  "instructedAmount": {
    "amount": {
      "value": 122.22,
      "currency": "CZK"
    }
  }
},

```

4.14.1.2.1 *transactionAmount (Transaction amount)*

JSON record: entryDetails.transactionDetails.amountDetails.instructedAmount.transactionAmount

Occurrence: [0..0]

Definition: Payment amount and currency

Application: A field not supported by the standard which, however, may be provided by any bank individually. It is used in cases where cumulated payments and Cashback are cleared.

Type: This message element consists of the following elements:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++++	amount	[1..1]	-	Amount definition

4.14.1.2.2 *counterValueAmount (Converted amount)*

JSON record: entryDetails.transactionDetails.amountDetails.instructedAmount.counterValueAmount

Occurrence: [0..1]

Definition: Payment amount and currency in the client account currency

Application: Transaction value after conversion of amount which was demanded by the client for transfer.

Type: This message element consists of the following elements:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++++	amount	[1..1]	±	Amount definition
+++++	currencyExchange	[0..1]	±	Exchange rate

JSON example of element:

```

"counterValueAmount": {
  "amount": {
    "currency": "EUR",
    "value": 86200.00
  },
  "currencyExchange": {
    "sourceCurrency": "EUR",
    "targetCurrency": "CZK",
    "exchangeRate": 27.01
  }
}

```

4.14.1.2.2.1 currencyExchange (Exchange rate)

JSON record:

entryDetails.transactionDetails.amountDetails.instructedAmount.counterValueAmount.currencyExchange

Occurrence: [0..1]

Definition: Information on currencies and Exchange rates used.

Application: Used in case that the payment currency and account currency are different, and an Exchange rate was used to make the transaction.

Type: This message element consists of the following elements:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++++	sourceCurrency	[1..1]	CurrencyCode	Amount definition

+++++	targetCurrency	[0..1]	CurrencyCode	Exchange rate
-------	----------------	--------	--------------	---------------

JSON example of element:

```

"currencyExchange": {
  "sourceCurrency": "EUR",
  "targetCurrency": "CZK",
  "exchangeRate": 27.01
}

```

4.14.1.2.2.1.1 sourceCurrency (Original currency)

JSON record:

entryDetails.transactionDetails.amountDetails.instructedAmount.counterValueAmount.currencyExchange.sourceCurrency

Occurrence: [1..1]

Definition: Client account currency

Application: Source/original currency/payer account currency for intrabank conversion payments.

Format type: CurrencyCode according to ISO 4217

JSON example of element:

```

"sourceCurrency": "EUR"

```

4.14.1.2.2.1.2 targetCurrency (Target currency)

JSON record:

entryDetails.transactionDetails.amountDetails.instructedAmount.counterValueAmount.currencyExchange.targetCurrency

Occurrence: [0..1]

Definition: Payment currency

Application: Final/target currency/payee account currency for intrabank conversion payments.

Format type: CurrencyCode according to ISO 4217

JSON example of element:

```
"targetCurrency": "CZK",
```

4.14.1.2.3 *proprietaryAmount (Cashback amount)*

JSON record: entryDetails.transactionDetails.amountDetails.instructedAmount.proprietaryAmount

Occurrence: [0..0]

Definition: Cash withdrawal amount through the Cashback.

Application: Only in case that the bank supports the field.

Type: This message element consists of the following elements:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++++	type	[0..0]	±	Transaction type
+++++	amount	[0..0]	Amount	Transaction amount

4.14.1.2.3.1 *type (Transaction type)*

JSON record: entryDetails.transactionDetails.amountDetails.instructedAmount.proprietaryAmount.type

Occurrence: [0..0]

Definition: Defines the transaction type

Application: Only for Cashback

Format type: constant „CASHBACK“

4.14.1.3 charges (Fees)

JSON record: entryDetails.transactionDetails.charges

Occurrence: [0..1]

Definition: Information on fees

Application: The field is defined only for foreign payments when from the original transaction amount, a fee may be deducted by the correspondent bank.

Type: This message element consists of the following elements:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
++++	bearer	[0..1]	-	Fee clearing side

4.14.1.3.1 bearer (Side for fees)

JSON record: entryDetails.transactionDetails.charges.bearer

Occurrence: [0..1]

Definition: Information on the side to which fees will be charged

Application: The side which bears the associated costs is defined by the code phrase:

DEBT – All fees paid by the payer

CRED – All fees paid by the payee

SHAR – Shared fees between the payer and payee.

SLEV – Fees required by the legislation or the payment scheme used, are applied

Format type: Code

4.14.1.4 relatedParties (Payer information)

JSON record: entryDetails.transactionDetails.relatedParties

Occurrence: [0..1]

Definition: Information on the payer, payer account and original payer and payee, payee account and final payee participating in the payment.

Application: The field is mandatory in the case of domestic, foreign and SEPA transfers

Type: This message element consists of the following elements:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++++	debtor	[0..1]	-	Payer
+++++	debtorAccount	[0..1]	-	Payer's account
+++++	ultimateDebtor	[0..1]	-	Original/actual payer
+++++	creditor	[0..1]	-	Payee
+++++	creditorAccount	[0..1]	-	Payee's account
+++++	ultimateCreditor	[0..1]	-	Original/actual payee
+++++	proprietary	[0..1]	-	Identification of the ATM on which the card transaction occurred

JSON examples of element:

```
"relatedParties": {
  "debtor": {
    "name": "RENWORTH s.r.o",
    "identification": {
      "organisationIdentification": {
        "other": [
          {
            "identification": "48135283",
            "schemeName": {
              "code": "1.2.203.48135283"
            }
          }
        ]
      }
    }
  }
}
```

```
}
```

```
"relatedParties": {  
  "debtor": {  
    "name": "RENWORTH s.r.o",  
    "identification": {  
      "organisationIdentification": {  
        "other": [  
          {  
            "identification": "48135283",  
            "schemeName": {  
              "proprietary": "partnership agreement"  
            }  
          }  
        ]  
      }  
    }  
  }  
}
```

4.14.1.4.1.1 type (Card transaction type)

JSON record: entryDetails.transactionDetails.relatedParties.proprietary.type

Occurrence: [0..0]

Definition: For card transactions, it is used for marking of whether it is own ATM or ATM of another.

Application: Only for card transactions. The field is not supported by the standard, but may be used by individual banks.

Format type: Max35Text

4.14.1.4.1.2 party (Operator)

JSON record: entryDetails.transactionDetails.relatedParties.proprietary.party

Occurrence: [0..0]

Definition: The field informs on the ATM owner/operator.

Application: Only for card transactions. The field is not supported by the standard, but may be used by individual banks.

Type: This message element consists of the following elements:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++++	name	[1..1]	Max140Text	Payer

4.14.1.5 relatedAgents (Payer and payee bank)

JSON record: entryDetails.transactionDetails.relatedAgents

Occurrence: [1..1]

Definition: Information on payer bank and payee bank.

Application: The element is present in payment transactions. In other cases (fees, interest, etc...) it makes no sense.

Type: This message element consists of the following elements:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++++	debtorAgent	[1..1]	±	Payer bank

JSON example of element:

```
"relatedAgents": {  
  "debtorAgent": {  
    "financialInstitutionIdentification": {
```

```

      "bic": "GIBACZPXXXX"
    }
  }
}

```

4.14.1.5.1 debtorAgent (Payer bank)

JSON record: entryDetails.transactionDetails.relatedAgents.debtorAgent

Occurrence: [1..1]

Definition: Information on payer bank.

Application: It is according to the payment direction, to be filled in for the counterparty. The element is present in payment transactions. In other cases (fees, interest, etc...) it makes no sense.

Type: This message element consists of the following elements:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++++	financialInstitutionIdentification	[1..1]	±	Financial institution identification

4.14.1.6 additionalTransactionInformation (Additional information)

JSON record: entryDetails.transactionDetails.additionalTransactionInformation

Occurrence: [0..1]

Definition: Additional information stated by the bank

Application: For information which is not defined in the standard structure. For instance, additional info for SEPA Direct Debit i.e. Creditor Identifier, Payment scheme, the order of SEPA collection, etc.

Format type: Max500Text

JSON example of element:

```

  "additionalTransactionInformation": "8201701069595 BIC:
GIBACZPXXXX; #71A# SHA ZALOHA DLE SMLOUVY O DODAVKACH,zaloha dle
smlouvy o dodavkach c. 45678/2017,VS0250117002/SS0000000000/KS0000SEPA
převod"

```

4.15 entryReference (Payment number)

JSON record: entryReference

Occurrence: [0..1]

Definition: Payment identification number

Application: A clear transaction identifier assigned by the bank.

Format type: Max35Text

JSON example of element:

```
"entryReference": "RB-4567813",
```

4.16 exchangeIdentification (Identifier assigned by third party)

JSON record: entryReference

Occurrence: [1..1]

Definition: Clear identification of request

Application: Identification is assigned by the third party; the value will be contained in the response to request.

Format type: Max18Text

JSON example of element:

```
"exchangeIdentification": "658576010faf0a23dc"
```


4.17 exchangeRateInformation (Contractual Exchange rate)

JSON record: exchangeRateInformation

Occurrence ALL: [0..0]

Definition: The element provides details about the exchange rate and Contract. The use and support of embedded element depends on the agreement with a specific bank.

Type: This message element consists of the following elements **ExchangeRateInformation1:**

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
++	exchangeRate	[0..0]	BaseOneRate	Agreed Exchange rate
++	rateType	[0..0]	ExchangeRate Type1Code	Type of Agreed Exchange rate
++	contractIdentification	[0..0]	Max35Text	Indicator of the use of Agreed Exchange rate

4.17.1 exchangeRate (Agreed Exchange rate)

JSON record: exchangeRateInformation.exchangeRate

Occurrence ALL: [0..0]

Definition: Detail of Agreed Exchange rate

Type: BaseOneRate

4.17.2 rateType (Type of Agreed Exchange rate)

JSON record: exchangeRateInformation.rateType

Occurrence ALL: [0..0]

Definition : Information on the type of Exchange rate used – AGREED, SALE, SPOT (agreed, sale, spot).

Type: ExchangeRateType1Code

4.17.3 *contractIdentification (Identifier of the use of Agreed Exchange rate)*

JSON record: exchangeRateInformation .contractIdentification

Occurrence ALL: [0..0]

Definition : Detail of Contract agreed with client for the execution of payments at the Agreed Exchange rate.

Type: Max35Text

4.18 *chargesAccount (Account for fees)*

JSON record: chargesAccount

Occurrence ALL: [0..0]

Definition: The account used to process transaction fees. The fee account is used if the fees are not to be charged to the debit of account marked as the payer's account. The fee account must be kept in the same bank as the payer's account.

Type: This message element consists of the following elements:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
++	identification	[0..0]	±	Identification of account no. for fees
++	currency	[0..0]	CurrencyCode – ISO 4217	Account currency for fees

4.18.1 *identification (Identification of account number for fees)*

JSON record: chargesAccount.identification

Occurrence ALL: [0..0]

Definition: Unique and unambiguous identification of fee account administered at the payer's bank.

Type: This message element consists of the following elements **AccountIdentification4CZ:**

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
-------	-----------------	------------	-------------	--------------

+++	IBAN	[0..0]	IBAN2007Identifier	Account no. for fees in the IBAN format
-----	------	--------	--------------------	---

4.18.1.1 IBAN (Account number for fees in the IBAN format)

JSON record: amount.equivalentAmount.curren chargesAccount.identification.iban

Occurrence ALL: [0..0]

Definition: International account number format used by financial institutions to uniquely and unambiguously identify a client account in accordance with the standard ISO 13616.

Data type: IBAN2007Identifier

Format for the Czech Rep.: [A-Z]{2,2}[0-9]{2,2}[a-zA-Z0-9]{1,20}, valid IBAN consists of all the following components: country code = CZ, check digit and account no. in local BBAN format, whereas BBAN consists of a four-digit numerical code of the bank, a six-digit prefix and a ten-digit account no. Initial zeros have no meaning.

Example: **CZ690710178124000000415**

4.18.2 currency (Fee account currency)

JSON record: chargesAccount.currency

Occurrence ALL: [0..0]

Definition: The currency in which the charge account is kept. If that currency does not match the charge account number, the payment order may be rejected.

Format type: [A-Z]{3,3} - CurrencyCode, ISO 4217

4.19 chargeBearer (Fee payer)

JSON record: chargeBearer

Occurrence:

TUZEM (Domestic payments): [0..0]

SEPA (SEPA payments): [0..0]

EHP (Foreign payments within EEA): [0..1]

NONEHP (Foreign payments outside EEA): [0..1]

Definition: Provides the party (parties) that will pay fees associated with the payment transaction processing.

TUZEM (Domestic payments): The code of the fee payer cannot be chosen (the payer always pays the fees of the payer's bank and the payee pays the fees of the payee's bank). When processing domestic payments within CNB clearing, the fee payer's code is not stated.

SEPA (SEPA payments): The code of the fee payer cannot be chosen (the payer always pays the fees of the payer's bank and the payee pays the fees of the payee's bank). In SEPA payment processing, the payer's bank will always complete the SLEV value.

EHP (Foreign payments within EEA): Permitted values are DEBT = OUR or SHAR = SHA. If no value is filled in, the payer's bank will fill in the SHAR value.

NONEHP (Foreign payments outside EEA): Permitted values are DEBT = OUR or CRED = BEN or SHAR = SHA. If no value is filled in, the payer's bank will fill in the SHAR value.

Data type: ChargeBearerType1Code

CODE	NAME	DEFINITION
DEBT	BorneByDebtor	All transaction charges are to be borne by the debtor.
CRED	BorneByCreditor	All transaction charges are to be borne by the creditor.
SHAR	Shared	In a credit transfer context, means that transaction charges on the sender side are to be borne by the debtor, transaction charges on the receiver side are to be borne by the creditor.
SLEV	FollowingServiceLevel	Charges are to be applied following the rules agreed in the service level and/or scheme.

4.20 intermediaryAgent1 (Intermediary bank 1)

JSON record: intermediaryAgent1

Occurrence All: [0..0]

Definition: Financial institution (Intermediary bank), through which money goes to the payee's bank.

Type: This message element consists of the following elements

BranchAndFinancialInstitutionIdentification4CZ:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
++	financialInstitutionIdentification	[0..0]	±	Financial institution identification

4.20.1 financialInstitutionIdentification (Financial institution identification)

JSON record: intermediaryAgent1.financialInstitutionIdentification

Occurrence All: [0..0]

Definition: Unique and unambiguous identification of the financial institution assigned by the international standard or using own identification scheme.

Type: This message element consists of the following elements **FinancialInstitutionIdentification7CZ**

4.20.1.1 bic (BIC / SWIFT code of bank)

JSON record: intermediaryAgent1.financialInstitutionIdentification.bic

Occurrence All: [0..0]

Definition: Bank Identifier Code. A code assigned to financial institutions by the Registration authority according to the international identification scheme as described in the latest version of the ISO 9362. The bank code in the format of BIC / SWIFT code has exactly 8 or exactly 11 alphanumeric characters.

Data type: BICIdentifier

4.20.1.2 clearingSystemMemberIdentification (Identification of clearing system participant)

JSON record: intermediaryAgent1.financialInstitutionIdentification.clearingSystemMemberIdentification

Occurrence All: [0..0]

Definition: Identification of the participant in the local clearing system

Type: This message element consists of the following elements **ClearingSystemMemberIdentification2**

4.20.1.2.1 clearingSystemIdentification (Identification of clearing system)

JSON record:

intermediaryAgent1.financialInstitutionIdentification.clearingSystemMemberIdentification.clearingSystemIdentification

Occurrence All: [0..0]

Definition: Identification of the clearing system

Type: This message element consists of the following elements **ClearingSystemIdentification2Choice:**

4.20.1.2.1.1 code (Code)**JSON record:**

intermediaryAgent1.financialInstitutionIdentification.clearingSystemMemberIdentification.clearingSystemIdentification.code

Occurrence All: [0..0]

Definition: A code identifying the local clearing system as specified in the external list of codes.

Data type: ExternalClearingSystemIdentification1Code

4.20.1.2.1.2 proprietary (Free format)**JSON record:**

intermediaryAgent1.financialInstitutionIdentification.clearingSystemMemberIdentification.clearingSystemIdentification.proprietary

Occurrence All: [0..0]

Definition: Identification of local clearing system, in free format.

Data type: Max35Text

4.20.1.2.2 memberIdentification (Participant's Identification code – clearing code)**JSON record:**

intermediaryAgent1.financialInstitutionIdentification.clearingSystemMemberIdentification.memberIdentification

Occurrence All: [0..0]

Definition: Identification code of the Participant or the bank's clearing code.

Data type: Max35Text

4.20.1.3 name (Name)

JSON record: intermediaryAgent1.financialInstitutionIdentification.name

Occurrence All: [0..0]

Definition: Name under which the party is known, and which is normally used to identify the party.

Data type: Max70Text

4.20.1.4 postalAddress (Postal address)

JSON record: intermediaryAgent1.financialInstitutionIdentification.postalAddress

Occurrence All: [0..0]

Definition: Information that localizes and identifies the specific address such as the postal address.

Type: This message element consists of the following elements **PostalAddress6CZ**

If for foreign payments it is agreed that this element should be stated, it is recommended to fill in the unstructured address form. It is recommended to use / fill in the Country field from the Country field and up to two address lines from the **Address Line** field. Typically, the first line shows the street and the Land registry number, the city and city's postal code in the second line.

4.20.1.4.1 streetName (Street)

JSON record: intermediaryAgent1.financialInstitutionIdentification.postalAddress.streetName

Occurrence All: [0..0]

Definition: Street name or section.

Data type: Max70Text

4.20.1.5 buildingNumber (Building number)

JSON record: intermediaryAgent1.financialInstitutionIdentification.postalAddress.buildingNumber

Occurrence All: [0..0]

Definition: Number that identifies the position of building in the street.

Data type: Max16Text

4.20.1.6 postCode (Postcode)

JSON record: intermediaryAgent1.financialInstitutionIdentification.postalAddress.postCode

Occurrence All: [0..0]

Definition: Identifier consisting of a group of letters and figures which are assigned to a postal address so that post may be sorted.

Data type: Max16Text

4.20.1.7 townName (Town)

JSON record: intermediaryAgent1.financialInstitutionIdentification.postalAddress.townName

Occurrence All: [0..0]

Definition: Name of the built-up area with defined boundaries and local self-administration.

Data type: Max35Text

4.20.1.8 country (Country)

JSON record: intermediaryAgent1.financialInstitutionIdentification.postalAddress.country

Occurrence All: [0..0]

Definition: A country with its own government

Data type: CountryCode, ISO 3166 (2 alphanumeric characters code version)

4.20.1.9 addressLine (Unstructured address record)

JSON record: intermediaryAgent1.financialInstitutionIdentification.postalAddress.addressLine

Occurrence All: [0..0]

Definition: Information that localizes and identifies the specific address as defined by postal services, presented in the free text format.

Data type: Max70Text

Maximum 2 lines with 70 characters per line

4.20.1.10 other (Another bank identification)

JSON record: intermediaryAgent1.financialInstitutionIdentification.other

Occurrence All: [0..0]

Definition: Unique agent identification assigned to the institution using the identification scheme

Type: This message element consists of the following elements **GenericFinancialIdentification1CZ**

4.20.1.10.1 identification (Local bank code)

JSON record: intermediaryAgent1.financialInstitutionIdentification.other.identification

Occurrence All: [0..0]

Definition: Local format of the bank code

Data type : Max35Text

4.21 instructionForNextAgent (Instruction for another bank)

JSON record: instructionForNextAgent

Occurrence All: [0..0]

Definition: Other information related to the payment order processing which is to be executed by another bank.

Type: Instruction code

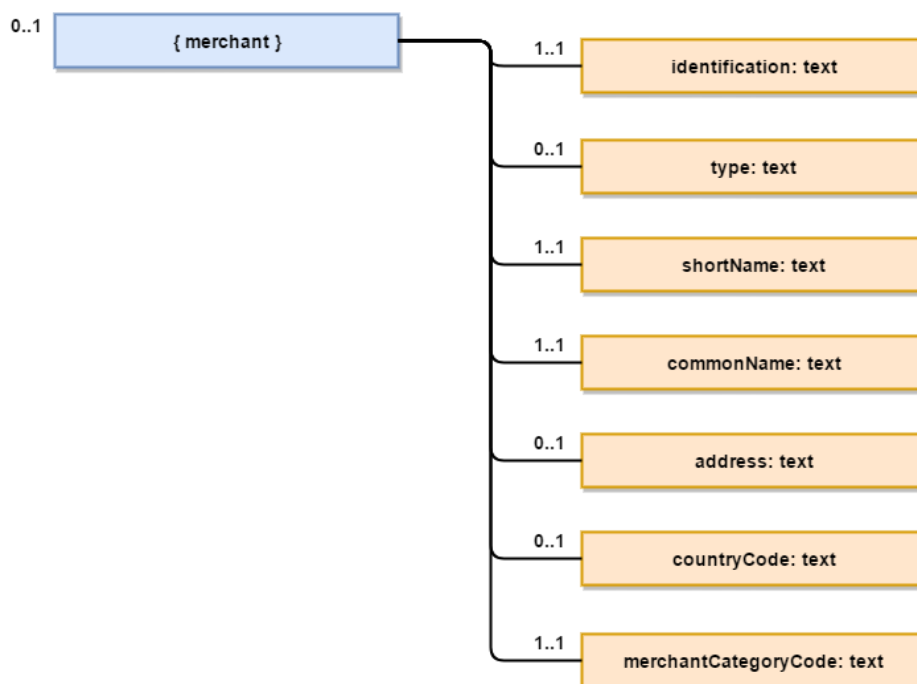
4.22 merchant (Merchant)

JSON record: merchant

Occurrence: [0..1]

Definition: A merchant executing a balance query

Application: In case that the third party and merchant are different entities.



Element **merchant** contains the following embedded elements:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION	LEVEL
++	identification	[1..1]	CISP	Max35Text	Merchant identification
++	type	[0..1]	CISP	Code	Merchant type
++	shortName	[1..1]	CISP	Max35Text	Merchant name
++	commonName	[1..1]	CISP	Max70Text	Merchant name as stated in the

					payment receipt
++	address	[0..1]	CISP	Max140Text	Merchant address
++	countryCode	[0..1]	CISP	CountryCode, ISO 3166 (2 alphanumeric characters code version)	Merchant country
++	merchantCategoryCode	[1..1]	CISP	Min3Max4Text, ISO 18245	Merchant code following the transaction type

JSON example of element:

```

"merchant": {
  "identification": "471 16 129",
  "shortName": "NEOLUXOR",
  "commonName": "Neoluxor s.r.o.",
  "address": "Hlavní 5, Praha 1",
  "countryCode": "CZ",
  "merchantCategoryCode": "5192"
},

```

4.22.1 identification (Identification)

JSON record: merchant.identification

Occurrence: [1..1]

Definition: Identification of an entity requesting a disposable balance

Application: Please specify the Reg. No. (IČO), in the case of an international payment, it is possible to use its variant of the country concerned.

Format type: Max35Text

JSON example of element:

```
"identification": "471 16 129"
```

4.22.2 *type (Entity type)*

JSON record: merchant.type

Occurrence: [0..1]

Definition: Type of entity executing a query for disposable balance

Format type: Code

CODE	NAME	DEFINITION
OPOI	OriginatingPOI	Point Of Interaction initiating the card payment transaction.
MERC	Merchant	Merchant providing goods and service in the card payment transaction.
ACCP	Acceptor	Card acceptor, party accepting the card and presenting transaction data to the acquirer.
ITAG	IntermediaryAgent	Party acting on behalf of other parties to process or forward data to other parties.
ACQR	Acquirer	Entity acquiring card transactions.
CISS	CardIssuer	Party that issues cards.
DLIS	Delegatelssuer	Party to whom the card issuer delegates to authorise card payment transactions.

4.22.3 *shortName (Entity name)*

JSON record: merchant.shortName

Occurrence: [1..1]

Definition: Name of entity executing a query for disposable balance – shortened form.

Format type: Max35Text

JSON example of element:

```
"shortName": "NEOLUXOR"
```

4.22.4 commonName (Entity name)

JSON record: merchant.commonName

Occurrence: [1..1]

Definition: Full name of entity as it will be stated in the payment receipt.

Format type: Max70Text

JSON example of element:

```
"commonName": "Neoluxor s.r.o.",
```

4.22.5 address (Address)

JSON record: merchant.address

Occurrence: [0..1]

Definition: Address of entity executing a query for disposable balance.

Application: If the address of the establishment is not the same as the entity address, the address of the establishment or the place where the disposable balance request was initiated, will be given. The address is given without the country name.

Format type: Max140Text

JSON example of element:

```
"address": "Hlavní 5, Praha 1",
```

4.22.6 *countryCode (Country code)*

JSON record: merchant.address

Occurrence: [0..1]

Definition: The country of the entity making the query for disposable balance in ISO format of 2 alphanumeric characters supported by SWIFT

Format type: CountryCode, ISO 3166 (2 alphanumeric characters code version)

JSON example of element:

```
"countryCode": "CZ"
```

4.22.7 *merchantCategoryCode (Category code)*

JSON record: merchant.merchantCategoryCode

Occurrence: [1..1]

Definition: ISO 18245 category code that applies to the type of service or goods according to the purpose for which the query for disposable balance is made.

Format type: Min3Max4Text, ISO 18245

JSON example of element:

```
"merchantCategoryCode": "5192"
```

4.23 paymentIdentification (Payment identification)

JSON record: paymentIdentification

Occurrence ALL: [1..1]

Definition: Set of elements used to identify the payment instruction.

Type: This message element consists of the following elements **PaymentIdentification1**:

LEVEL	MESSAGE ELEMENT	FORMAT TYPE	PRESENTATION
++	instructionIdentification	Max35Text	Instruction Identification
++	endToEndIdentification	Max35Text	End To End Identification
++	transactionIdentification	Max35Text	Transaction identification

JSON example of element:

```
"paymentIdentification": {  
  "instructionIdentification": "NOTPROVIDED",  
  "transactionIdentification": "048885570000001020045"  
}
```

4.23.1 instructionIdentification (Instruction identification)

JSON record: paymentIdentification.instructionIdentification

Occurrence ALL: [1..1]

Definition: Unique identification assigned by PISP that uniquely / unambiguously identifies the instruction. Instruction identification is a Point-to-Point reference that can be used between the instructor and the instructed party related to this one (individual) instruction. It may appear in several messages related to the instruction.

Format type: Max35Text

JSON example of element:

```
"instructionIdentification": "NOTPROVIDED",
```

4.23.2 *endToEndIdentification (End To End identification)*

JSON record: paymentIdentification.endToEndIdentification

Occurrence:

TUZEM (Domestic payments): [0..0]

SEPA (SEPA payments): [1..1]

EHP (Foreign payments within EEA): [0..0]

NONEHP (Foreign payments outside EEA): [0..0]

Definition: Identification agreed between the payer and the payee. If not filled in by the payer, then PISP will fill in „NOTPROVIDED“.

Format type: Max35Text

JSON example of element:

```
"endToEndIdentification": "048885570000001020045",
```

4.23.3 *transactionIdentification (Transaction identification)*

JSON record: paymentIdentification.transactionIdentification

Occurrence request ALL: [0..0]

Occurrence response ALL: [1..1]

Definition: The unique reference of the payer's bank. This identification is returned by the provider who maintains the payer's account in response to the received request for a new payment and / or a payment status query.

Format type: Max35Text

JSON example of element:

```
"transactionIdentification": "048885570000001020045"
```

4.24 paymentTypeInformation (Information on payment type)

JSON record: paymentTypeInformation

Occurrence ALL: [0..1] when used in a request; [1..1] when used in a response

Definition: A set of elements used to further determine the transaction type. It is used to determine the speed and method of payment processing.

Type: This message element consists of the following elements **PaymentTypeInformation19:**

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
++	instructionPriority	[0..1]	Code	Instruction priority
++	serviceLevel	[0..0] when used in a request; [1..1] when used in a response	±	Service level
++	categoryPurpose	[0..0]	±	Payment purpose category

JSON example of element:

```
"paymentTypeInformation": {  
  "instructionPriority": "NORM",  
  "serviceLevel": {  
    "code": "DMCT"  
  }  
},
```

4.24.1 *instructionPriority* (Instruction priority)

JSON record: paymentTypeInformation.instructionPriority

Occurrence ALL: [0..1]

Definition: If the element is not filled in, the payer's provider assigns a NORM value and makes a payment by default. If a "HIGH" value is indicated, the payer's provider will make the payment as priority / express / urgent, usually at D + 0 foreign currency. If the "NORM" is specified, they will make the payment by default, i.e. D + 1 foreign currency, where D, the date of debiting funds from the payer's account. The deadlines for making standard and priority payments may vary depending on the type of payment and are governed by the terms of each payer's conditions.

If the "INST" value is stated, the payer's provider executes the payment as Instant according to the applicable rule.

Data type: Priority2Code

JSON example of element:

```
"instructionPriority": "NORM"
```

JSON example of element for Instant payment:

```
"instructionPriority": "INST"
```

Error example body:

```
{
  "errors": [
    {
      "error": "NO_PART",
      "parameters": {
        "ALLOWED_PRIORITY": "NORM"
      },
      "scope": "paymentTypeInformation.instructionPriority"
    }
  ]
}
```

```
}  
]  
}
```

4.24.2 serviceLevel (Service level)

JSON record: paymentTypeInformation.serviceLevel

Occurrence ALL: [0..0] when used in a request; [1..1] when used in a response

Definition: An agreement or rules according to which the transaction is to be processed.

Data type: This message element consists of the following elements **ServiceLevel8CZ**

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++	Code	[0..0] when used in a request; [1..1] when used in a response	Code	Code

JSON example of element:

```
"serviceLevel": {  
  "code": "DMCT"  
}
```

4.24.2.1 code (Code)

JSON record: paymentTypeInformation.serviceLevel.code

Occurrence ALL: [0..0] when used in a request; [1..1] when used in a response

Definition: An agreement or rules according to which the transaction is to be processed.

CODE	DESCRIPTION
DMCT	[DoMestic Credit Transfer] Domestic payment
ESCT	[SEPA Credit Transfer] – SEPA payment
XBCT	[Cross-Border Credit Transfer] – Cross border payment
EXCT	[EEA X-border Credit Transfers] – Cross border payment within the EEP
NXCT	[Non-EEA X-border Credit Transfers] – Cross border payment outside the EEP

Data type: ExternalServiceLevel1Code

JSON example of element:

```

"serviceLevel": {
  "code": "DMCT"
}

```

4.24.3 *categoryPurpose (Payment purpose category)*

JSON record: paymentTypeInformation.categoryPurpose

Occurrence ALL: [0..0]

Definition: Instruction code to the payee's bank or payer's bank for an agreed specific payment processing method.

Data type: This message element consists of the following elements **CategoryPurpose1Choice**

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
+++	Code	[0..0]	Code	Code
+++	Proprietary	[0..0]	Max35Text	Free format

4.24.3.1 code (Code)

JSON record: paymentTypeInformation.categoryPurpose.code

Occurrence ALL: [0..0]

Definition: An agreement or rules according to which the transaction is to be processed.

Data type: ExternalCategoryPurpose1Code

4.24.3.2 proprietary (Free format)

JSON record: paymentTypeInformation.categoryPurpose.proprietary

Occurrence ALL: [0..0]

Definition: Payment purpose category, in free format..

Data type: Max35Text

4.25 initiatingParty (Initiating Party)

JSON record: initiatingParty

Occurrence ALL: [0..1]

Definition: TPP identification element – the name of the original third party

Data type: This message element consists of the following elements

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
++	Name	[0..1]	Max35Text	Free format
++	PostalAddress	[0..1]	Max35Text	Free format

JSON example of element:

```
"initiatingParty": {  
  "name": "Alza, a.s.",  
  "postalAddress": "Jateční 33a, 170 00 Praha-Holešovice-Praha 7"  
},
```

4.25.1 name

JSON record: initiatingParty.name

Occurrence ALL: [0..1]

Definition: Name of TPP

4.25.2 postalAddress

JSON record: initiatingParty.postalAddress

Occurrence ALL: [0..1]

Definition: Postal address of TPP

4.26 purpose (Payment purpose)

JSON record: purpose

Occurrence All: [0..0]

Definition: Normalized payment reason code – optional field. To fill in payment information, for example, what kind of goods / services are paid. More details in ISO codes, at this link: www.ISO20022.org.

Type: This message element consists of the following elements **Purpose2Choice**:

LEVEL	OR	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
++	{Or	code	[0..0]	Code	Code
++	Or}	proprietary	[0..0]	Max35Text	Free format

JSON example of element:

```
"purpose": {  
  "proprietary": "PLATBA ZA SLUŽBY"  
},
```

4.26.1 code (Code)

JSON record: purpose.code

Occurrence All: [0..0]

Definition: Payment purpose as stated in the external list of payment purpose codes.

Data type: ExternalPurpose1Code

4.26.2 proprietary (Free format)

JSON record: purpose.proprietary

Occurrence All: [0..0]

Definition: Payment purpose, in free format.

Data type: Max35Text

JSON example of element:

```
"proprietary": "PLATBA ZA SLUŽBY"
```

4.27 remittanceInformation (Payment information)

JSON record: remittanceInformation

Occurrence All: [0..1]

Definition: Information that allows match (i.e. pairing, reconciliation) of payments with which the payment should be settled, such as commercial invoices in the receivables system. Use the Message for payee, e.g., to forward a variable, specific, and constant symbol.

Type: This message element consists of the following elements **RemittanceInformation5CZ:**

LEVEL	MESSAGE ELEMENT	FORMAT TYPE	PRESENTATION
++	unstructured	Max140Text	Unstructured message for payee
++	structured	±	Structured message for payee – variable, specific, and constant symbol
++	additionalRemittanceInformation	Max140Text	Additional free text information of up to 140 characters.

JSON example of element:

```

"remittanceInformation": {
  "unstructured": "\`",
  "structured": {
    "creditorReferenceInformation": {
      "reference": "VS:123456\`,`KS:456789\`,`SS:879213546"
    }
  }
},

```

4.27.1 unstructured (Unstructured message for payee)

JSON record: remittanceInformation.unstructured

Occurrence All: [0..1]

Definition: If you require a variable, specific, and / or constant symbol to be forwarded to the payee, specify the unstructured information at the beginning in the form of VS/max.10 digits/SS/max.10 digits/KS/max.10 digits.

Data type: Max140Text

JSON example of element:

```

"unstructured": "\`"

```


4.27.2 structured (Structured message for payee – variable, specific and constant symbol)

JSON record: remittanceInformation.structured

Occurrence only TUZEM (Domestic payments): [0..3]

Definition: Information allowing the matching of the variable, specific or constant symbol for the items with which the transfer should be settled. Structured Remittance Information can be used to fill in maximum one variable symbol, one specific symbol, or one constant symbol.

Type: This message element consists of the following elements **StructuredRemittanceInformation7CZ:**

LEVEL	MESSAGE ELEMENT	FORMAT TYPE	PRESENTATION
+++	creditorReferenceInformation	±	Information on payee reference

JSON example of element:

```
"structured": {
  "creditorReferenceInformation": {
    "reference": "VS:123456\", \"KS:456789\", \"SS:879213546"
  }
}
```

4.27.2.1 creditorReferenceInformation (Information on payee reference)

JSON record: remittanceInformation.structured.creditorReferenceInformation

Occurrence only TUZEM (Domestic payments): [0..1]

Definition: Reference information provided by the payee that allows identification of underlying documents (e.g. invoice).

Type: This message element consists of the following elements **CreditorReferenceInformation2CZ:**

LEVEL	MESSAGE ELEMENT	FORMAT TYPE	PRESENTATION
++++	reference	Max35Text VS – variable, SS – specific, KS –	Value of VS, SS, KS

		constant symbol. Each of the symbols as separate structured data.	
--	--	--	--

JSON example of element:

```

"creditorReferenceInformation": {
  "reference": "VS:123456\","KS:456789\","SS:879213546"
}

```

4.27.2.1.1 reference (Value of VS, SS, KS)

JSON record: remittanceInformation.structured.creditorReferenceInformation.reference

Occurrence only TUZEM (Domestic payments): [0..1]

Definition: Information allowing to pass on information about the value of a variable symbol, a specific symbol, or a constant symbol. The value of variable symbol is recorded as VS:max.10 digits (e.g. VS:3451859072). The value of specific symbol is recorded as SS:max.10 digits (e.g. SS:8451201274). The value of constant symbol is recorded as KS:max.10 characters (e.g. KS:0308).

Data type: Max35Text

JSON example of element:

```

"reference": "VS:123456\","KS:456789\","SS:879213546"

```

4.27.3 additionalRemittanceInformation (Unstructured message for payer)

JSON record: remittanceInformation.additionalRemittanceInformation

Occurrence All: [0..1]

Definition: Unstructured message for payer. Additional free text information of up to 140 characters.

Data type: Max140Text

JSON example of element:

```
"additionalRemittanceInformation":  
    "Platba za MŠ ..."
```

4.28 requestedExecutionDate (Required payment execution date)

JSON record: interbankSettlementDate/requestedExecutionDate

Occurrence ALL: [0..1]

Definition: The date on which the initiating party requests payment processing by a settlement agent. On this date, funds will be debited from the payer's account if the agreed terms are met. The bank terms pertaining to the execution date may vary, for example, depending on the business hours, the work day specification for the type of payment, the number of days to track the available balance, the number of days for future maturity, the date older than the current day, etc. If the requested execution date is not filled in, the payer's bank will debit funds from the payer's account immediately after meeting the agreed terms.

Data type: ISODate

JSON example of element:

```
"requestedExecutionDate": "2017-01-31"
```

4.29 reversalIndicator (Cancellation)

JSON record: reversalIndicator

Occurrence: [0..1]

Definition: Indication of whether it is cancellation.

Application: Only acquires the values ""true"", or ""false"":

true: It is cancellation

false: It is not cancellation

Format type: TrueFalseIndicator

JSON example of element:

```
"reversalIndicator": false
```

4.30 standingOrder (Recurring payment instruction specification)

JSON record: standingOrder

Occurrence: [1..1]

Definition: Structure describing the parameters of the standing orders.

Type: This message element consists of the following elements:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
++	alias	[0..1]	ALL	Max250Text	Description resp. the user-defined standing order payment name.
++	execution	[1..1]	ALL		Features of execution a standing order for payment.
+++	mode	[1..1]	ALL	Max35Text	The execution mode defines when or how standing order will be cancelled, processed the last time.
+++	modeDue	[1..1]	ALL	Max35Text	The execution due mode defines how the date when order should be executed is specified.
+++	interval	[1..1]	ALL	Max10Text	Execution interval defines how often order is executed.

+++	intervalDue	[1..1]	ALL	Number	Value represents order number of the day within particular period when the standing order will be regularly executed.
++	validity	[0..1]	ALL		Elements restricting the validity of the standing order.
+++	lastExecutionDate	[0..1]	ALL	ISODate	Date when the last order will be processed. Null value responds to ending by the user.
+++	maxIterations	[0..1]	ALL	Number	Maximum number of iterations - processing of the standing order.
+++	maxAmount	[0..1]	ALL	Amount	Maximum amount to be transferred using the standing order.
++++	value	[1..1]	ALL	Amount	Maximum amount of the transfer
++++	currency	[1..1]	ALL	CurrencyCode	Transfer Currency of the maximum amount
++	exceptions	[0..1]	ALL		Elements defining exceptions to executing a standing order.
+++	stoppages	[0..1]	ALL	Array of Max20Text	List of months where there is no payment.
+++	breaks	[0..1]	ALL	Array of Intervals	Break periods.
++++	validFromDate	[1..1]	ALL	ISODate	Start date of one break period. Standing order will not be processed from this date.
++++	validToDate	[1..1]	ALL	ISODate	End date of one break period. Standing order will not be processed to this date.

JSON example of element:

```
"standingOrder": {
  "alias": "Payments for lunch - kindergarten Michal",
```

```

"execution": {
  "interval": "MONTHLY",
  "intervalDue": "25",
  "mode": "MAX_AMOUNT_EXCEEDED",
  "modeDue": "DUE_OR_BEFORE_DAY_OF_MONTH"
},
"exceptions": {
  "stoppages": ["JULY", "AUGUST"],
  "breaks": [
    {
      "validFromDate": "2018-11-01",
      "validToDate": "2019-01-30"
    },
    {
      "validFromDate": "2019-05-01",
      "validToDate": "2019-06-15"
    }
  ]
},
"validity": {
  "lastExecutionDate": null,
  "maxAmount": {
    "value": 89000.00,
    "currency": "CZK"
  }
}
}

```

4.30.1 execution (Features of execution)

JSON record: standingOrder.execution

Occurrence: [1..1]

Definition: Features of execution a standing order for payment.

Type: This message element consists of the following elements:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
+++	mode	[1..1]	ALL	Max35Text	The execution mode defines when or how standing order will be cancelled, processed the last time.
+++	modeDue	[1..1]	ALL	Max35Text	The execution due mode defines how the date when order should be executed is specified.
+++	interval	[1..1]	ALL	Max10Text	Execution interval defines how often order is executed.
+++	intervalDue	[1..1]	ALL	Number	Value represents order number of the day within particular period when the standing order will be regularly executed.

4.30.1.1 mode (Standing order execution mode)

JSON record: standingOrder.execution.mode

Occurrence: [1..1]

Format type: Max35Text

Definition: The execution mode defines when or how standing order will be cancelled, processed the last time.

Possible values:

CODE	DESCRIPTION
UNTIL_DATE	Standing order is valid until specific date - field lastExecutionDat .
UNTIL_CANCELLATION	Standing order is valid forever and must be cancelled by client.

AFTER_MAX_ITERATION_EXCEEDED	Certain count of executions is specified - field <code>maxIterations</code>
MAX_AMOUNT_EXCEEDED	Maximum amount which can be transferred for this order is specified, if next iteration would exceed this amount it is not executed - field <code>maxAmount</code>

JSON example of element UNTIL_CANCELLATION:

```
"mode": "UNTIL_CANCELLATION",
```

JSON example of element UNTIL_DATE:

```
"lastExecutionDate": "2019-09-30",
...
"mode": "UNTIL_DATE",
```

4.30.1.2 modeDue (Standing order execution due mode specification)

JSON record: `standingOrder.execution.modeDue`

Occurrence: [1..1]

Format type: Max35Text

Definition: The execution due mode defines how the date when order should be executed is specified.

Possible values:

<i>CODE</i>	<i>DESCRIPTION</i>
DUE_DAY_OF_MONTH	Specific number of day in the month is defined.
DUE_OR_BEFORE_DAY_OF_MONTH	Is defined as a specific day of the month or the previous day if the day falls on a non-banking day.
DUE_OR_NEXT_DAY_OF_MONTH	Order is executed on last day of particular month

JSON example of element:

```
"modeDue": "DUE_DAY_OF_MONTH",
```

4.30.1.3 interval (Standing order execution interval)

JSON record: standingOrder.execution.interval

Occurrence: [1..1]

Format type: Max10Text

Definition: Execution interval defines how often order is executed.

Possible values:

<i>CODE</i>	<i>DESCRIPTION</i>
DAILY	Once per day
WEEKLY	Once a week
MONTHLY	Once a month
BI_MONTHLY	Once every two months
QUARTERLY	Is done once every quarter of a year
HALFYEARLY	Will be done once every six months
YEARLY	Is done once a year
IRREGULAR	It is performed irregularly

JSON example of element:

```
"interval": "IRREGULAR",
```

4.30.1.4 intervalDue (Standing order due interval)

JSON record: standingOrder.execution.intervalDue

Occurrence: [1..1]

Format type: Number

Definition: Value represents order number of the day within particular period when the standing order will be regularly executed.

Possible values:

VALUE	DESCRIPTION
1 - 7	For WEEKLY execution interval (executionInterval)
1 - 28	For MONTHLY execution interval (executionInterval)
1 - 2	For BI_MONTHLY execution interval (executionInterval)
1 - 3	For QUARTERLY execution interval (executionInterval)
1 - 6	For HALFYEARLY execution interval (executionInterval)
1 - 12	For YEARLY execution interval (executionInterval)

JSON example of element in case of MONTHLY execution interval:

```
"interval": "MONTHLY",  
...  
"intervalDue": 15,
```

4.30.2 validity (restricting the standing order validity)

JSON record: standingOrder.validity

Occurrence: [0..1]

Definition: Elements restricting the validity of the standing order.

Type: This message element consists of the following elements:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
+++	lastExecutionDate	[0..1]	ALL	ISODate	Date when the last order will be processed. Null value responds to ending by the user.
+++	maxIterations	[0..1]	ALL	Number	Maximum number of iterations - processing of the standing order.
+++	maxAmount	[0..1]	ALL	Amount	Maximum amount to be transferred using the standing order.
++++	value	[1..1]	ALL	Amount	Maximum amount of the transfer
++++	currency	[1..1]	ALL	CurrencyCode	Transfer Currency of the maximum amount

4.30.2.1 lastExecutionDate (Standing order last execution date)

JSON record: standingOrder.validity.lastExecutionDate

Occurrence: [0..1]

Definition: Date when the last order will be processed. Null value responds to ending by the user.

Format type: ISODate

An JSON example when a standing order is to be terminated by a user in the future:

```
"lastExecutionDate": null,
```

An JSON example when a standing order is to be terminated on a certain date:

```
"lastExecutionDate": "2023-12-31T23:59:59.999Z",
```

```
"lastExecutionDate": "2019-09-30",
```

4.30.2.2 maxIterations (Standing order number of iterations)

JSON record: standingOrder.validity.maxIterations

Occurrence: [0..1]

Format type: Number

Definition: Maximum number of iterations - processing of the standing order. Only applicable in combination with **executionMode AFTER_MAX_ITERATION_EXCEEDED**.

JSON example of element:

```
"mode": "AFTER_MAX_ITERATION_EXCEEDED",  
...  
"maxIterations": 50,
```

4.30.2.3 maxAmount (Standing order maximum cumulative value)

JSON record: standingOrder.validity.maxAmount

Occurrence: [0..1]

Format type: Object

Definition: Maximum amount to be transferred using the standing order. Only applicable in combination with **executionMode AFTER_MAX_AMOUNT_EXCEEDED**.

This message element consists of the following elements:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
++++	value	[1..1]	ALL	Amount	Maximum amount of the transfer
++++	currency	[1..1]	ALL	CurrencyCode	Transfer Currency of the maximum amount

JSON example of element:

```
"mode": "AFTER_MAX_AMOUNT_EXCEEDED",  
...  
"maxAmount": {  
  "value": 500000.10,  
  "currency": "CZK"  
}
```

4.30.3 exceptions (exceptions to executing)

JSON record: standingOrder.exceptions

Occurrence: [0..1]

Definition: Elements defining exceptions to executing a standing order.

Type: This message element consists of the following elements:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
+++	stoppages	[0..1]	ALL	Array of Max20Text	List of months where there is no payment.
+++	breaks	[0..1]	ALL	Array of Intervals	Break periods.
++++	validFromDate	[1..1]	ALL	ISODate	Start date of one break period. Standing order will not be processed from this date.
++++	validToDate	[1..1]	ALL	ISODate	End date of one break period. Standing order will not be processed to this date.

4.30.3.1 stoppages (Standing order monthly stoppages)

JSON record: standingOrder.exceptions.stoppages

Occurrence: [0..1]

Format type: Array of Max20Text

Definition: List of months where there is no payment. Possible values: JANUARY, FEBRUARY, MARCH, APRIL, MAY, JUNE, JULY, AUGUST, SEPTEMBER, OCTOBER, NOVEMBER, DECEMBER

JSON example of element:

```
"stoppages": [ "JULY", "AUGUST" ],
```

4.30.3.2 breaks (Standing order breaks intervals)

JSON record: standingOrder.exceptions.breaks

Occurrence: [0..1]

Format type: Array of Intervals

Definition: Break periods for standing order.

This message element consists of the following elements:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
+++	validFromDate	[1..1]	ALL	ISODate	Start date of one break period. Standing order will not be processed from this date.
+++	validToDate	[1..1]	ALL	ISODate	End date of one break period. Standing order will not be processed to this date.

JSON example of element for two intervals definition:

```
"breaks": [  
  {  
    "validFromDate": "2018-11-01",  
    "validToDate": "2019-01-30"  
  },  
  {  
    "validFromDate": "2019-05-01",  
    "validToDate": "2019-06-15"  
  }  
],
```

4.30.3.2.1 *validFromDate (Standing order starting date of interval)*

JSON record: standingOrder.exceptions.breaks[0..n].validFromDate

Occurrence: [0..1]

Format type: ISODate

Definition: Start date of one break period. Standing order will not be processed from this date.

JSON example of element for two intervals definition:

```
"validFromDate": "2018-11-01",
```

4.30.3.2.2 *validToDate (Standing order ending date of interval)*

JSON record: standingOrder.exceptions.breaks[0..n].validToDate

Occurrence: [0..1]

Format type: ISODate

Definition: End date of one break period. Standing order will not be processed to this date.

JSON example of element for two intervals definition:

```
"validToDate": "2018-11-28",
```

4.31 status (Status)

JSON record: status

Occurrence: [1..1]

Definition: Item status in the account from the point of view of the bank

Application: In the statement, only posted items will be shown, with the constant BOOK, or blocked items, with the constant PDNG.

Format type: Code

JSON example of element:

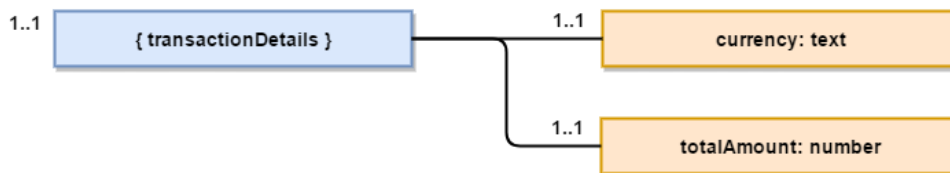
```
"status": "BOOK",
```

4.32 transactionDetails (Transaction details)

JSON record: transactionDetails

Occurrence: [1..1]

Definition: Transaction details



Element **transactionDetails** contains the following embedded elements:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PAYMENT TYPE	FORMAT TYPE	PRESENTATION
++	currency	[1..1]	CISP	CurrencyCode, ISO 4217	Currency of query for balance
++	totalAmount	[1..1]	CISP	Amount	Amount of query for balance

JSON example of element:

```

"transactionDetails": {
  "currency": "CZK",
  "totalAmount": 10050.15
}
  
```

4.32.1 currency (Currency)

JSON record: transactionDetails.currency

Occurrence: [1..1]

Definition: The currency of the amount that is verified by querying the available balance.

Format type: CurrencyCode, ISO 4217

JSON example of element:

```

"currency": "CZK"
  
```

4.32.2 totalAmount (Total amount)

JSON record: transactionDetails.totalAmount

Occurrence: [1...1]

Definition: Total amount that is verified by querying the available balance.

Format type: Amount

JSON example of element:

```
"currency": "CZK"
```

4.33 ultimateCreditor (Final payee)

JSON record: ultimateCreditor

Occurrence only SEPA (SEPA payments): [0..1]

Definition: Final party to whom the financial amount is due.

Type: This message element consists of the following elements **PartyIdentification32CZ1**:

LEVEL	MESSAGE ELEMENT	OCCURRENCE	FORMAT TYPE	PRESENTATION
++	Name	[0..1]	Max70Text	Name of final payee
++	Postal Address	[0..1]	±	Postal address of final payee
++	Identification	[0..1]	±	Identification of final payee

JSON example of element:

```

"ultimateCreditor": {
  "name": "Franz Schubert",
  "postalAddress": {
    "streetName": "Dianagasse",
    "buildingNumber": "6",
    "postCode": "1030",
    "townName": "Wiena",
    "country": "AT"
  },
  "identification": {
    "privateIdentification": {
      "other": [
        {
          "identification": "57842578842",
          "schemeName": {
            "proprietary": "passport",
            "issuer": "WIENA"
          }
        }
      ]
    }
  }
}

```

4.33.1 name (Name)

JSON record: ultimateCreditor.name

Occurrence only SEPA (SEPA payments): [0..1]

Definition: The name under which the party is known, and which is commonly used to identify that party.

Data type: Max70Text

JSON example of element:

```
"name": "Franz Schubert"
```

4.33.2 *postalAddress (Postal address)*

JSON record: ultimateCreditor.postalAddress

Occurrence only SEPA (SEPA payments): [0..1]

Definition: Information that locates and identifies a specific address as a postal address.

Type: This message element consists of the following elements **PostalAddress6CZ**

For SEPA payments, it is recommended to fill in the unstructured address form. It is recommended to use / fill in the Country field from the Country field and up to two lines of address from the Address Line field. Typically, the first line shows the street and the Land registry number, and the second line the town and the postal code.

JSON example of element:

```
"postalAddress": {  
  "streetName": "Dianagasse",  
  "buildingNumber": "6",  
  "postCode": "1030",  
  "townName": "Wiena",  
  "country": "AT"  
}
```

4.33.2.1 streetName (Street)

JSON record: ultimateCreditor.postalAddress.streetName

Occurrence: only SEPA payments: [0..1]

Definition: Name of street or section.

Data type: Max70Text

JSON example of element:

```
"streetName": "Dianagasse",
```

4.33.2.2 buildingNumber (Building number)

JSON record: ultimateCreditor.postalAddress.buildingNumber

Occurrence: only SEPA payments: [0..1]

Definition: Number identifying the position of the building in the street.

Data type: Max16Text

JSON example of element:

```
"buildingNumber": "6",
```

4.33.2.3 postCode (Postcode)

JSON record: ultimateCreditor.postalAddress.postCode

Occurrence: only SEPA payments: [0..1]

Definition: Identifier consisting of a group of letters and figures which are assigned to a postal address so that post may be sorted.

Data type: Max16Text

JSON example of element:

```
"postCode": "1030",
```

4.33.2.4 townName (Town)

JSON record: ultimateCreditor.postalAddress.townName

Occurrence: only SEPA payments: [0..1]

Definition: Name of the built-up area with defined boundaries and local self-administration.

Data type: Max35Text

JSON example of element:

```
"townName": "Wiena",
```

4.33.2.5 country (Country)

JSON record: ultimateCreditor.postalAddress.country

Occurrence: only SEPA payments: [0..1]

Definition: A country with its own government

Data type: CountryCode

Data format: [A-Z]{2,2}

Rule: Country code is checked against the list of country names under the ISO 3166 (2 alphanumeric characters code version).

JSON example of element:

```
"country": "AT"
```

4.33.2.6 addressLine (Unstructured address record)

JSON record: ultimateCreditor.postalAddress.addressLine

Occurrence: only SEPA payments: [0..2]

Definition: Information that localizes and identifies the specific address as defined by postal services, presented in the free text format.

Data type: Max70Text

Maximum 2 lines with 70 characters per line

4.33.3 identification (Identification)

JSON record: ultimateCreditor.identification

Occurrence: only SEPA payments: [0..1]

Definition: A unique and unambiguous identification of the party.

Type: This message element consists of the following elements **Party6Choice**:

LEVEL	OR	MESSAGE ELEMENT	OCCURRENCE	PRESENTATION
+++	{Or	Organisation Identification	[1..1]	Identification of organization
+++	Or}	Private Identification	[1..1]	Identification of private individual

JSON example of element:

```
"identification": {
  "privateIdentification": {
    "other": [
      {
        "identification": "57842578842",
        "schemeName": {
          "proprietary": "passport",
          "issuer": "WIENA"
        }
      }
    ]
  }
}
```

```

    }
  ]
}
}

```

4.34 ultimateDebtor (Original payee)

JSON record: ultimateDebtor

Occurrence: only SEPA payments: [0..1]

Definition: The final debtor paying through the payer.

Type: This message element consists of the following elements (for SEPA payments):

LEVEL	MESSAGE ELEMENT	FORMAT TYPE	PRESENTATION
++	Name	Max70Text	Name of original payee
++	Postal Address	±	Postal address of original payee
++	Identification	±	Identification

JSON example of element:

```

"ultimateDebtor": {
  "name": "Karel Novák",
  "postalAddress": {
    "streetName": "Vodičkova",
    "buildingNumber": "12",
    "postCode": "12000",
    "townName": "Praha 2",
    "country": "CZ"
  },
  "identification": {
    "privateIdentification": {
      "other": [

```



```
{
  "identification": "57842578842",
  "schemeName": {
    "proprietary": "passport",
    "issuer": "WIENA"
  }
}
]
```

4.34.1 *name (Name of original payee)*

JSON record: ultimateDebtor.name

Occurrence: only SEPA payments: [0..1]

Definition: The name under which the party is known, and which is commonly used to identify that party.

Data type: Max70Text

JSON example of element:

```
"name": "Karel Novák"
```

4.34.2 *postalAddress (Postal address of original payer)*

JSON record: ultimateDebtor.postalAddress

Occurrence: only SEPA payments: [0..1]

Definition: Information that locates and identifies a specific address as a postal address.

Type: This message element consists of the following elements **PostalAddress6CZ**.

For SEPA payments, it is recommended to fill in the unstructured address form. It is recommended to use / fill in the Country field from the Country field and up to two lines of address from the Address Line field. Typically, the first line shows the street and the Land registry number, and the second line the town and the postal code.

JSON example of element:

```
"postalAddress": {  
  "streetName": "Vodičkova",  
  "buildingNumber": "12",  
  "postCode": "12000",  
  "townName": "Praha 2",  
  "country": "CZ"  
}
```

4.34.2.1 streetName (Street)

JSON record: ultimateDebtor.postalAddress.streetName

Occurrence: only SEPA payments: [0..1]

Definition: Name of street or section.

Data type: Max70Text

JSON example of element:

```
"streetName": "Vodičkova"
```

4.34.2.2 buildingNumber (Building number)

JSON record: ultimateDebtor.postalAddress.buildingNumber

Occurrence: only SEPA payments: [0..1]

Definition: Number identifying the position of the building in the street.

Data type: Max16Text

JSON example of element:

```
"buildingNumber": "12"
```

4.34.2.3 postCode (Postcode)

JSON record: ultimateDebtor.postalAddress.postCode

Occurrence: only SEPA payments: [0..1]

Definition: Identifier consisting of a group of letters and figures which are assigned to a postal address so that post may be sorted.

Data type: Max16Text

JSON example of element:

```
"postCode": "12000"
```

4.34.2.4 townName (Town)

JSON record: ultimateDebtor.postalAddress.townName

Occurrence: only SEPA payments: [0..1]

Definition: Name of the built-up area with defined boundaries and local self-administration.

Data type: Max35Text

JSON example of element:

```
"townName": "Praha 2"
```

4.34.2.5 country (Country)

JSON record: ultimateDebtor.postalAddress.country

Occurrence: only SEPA payments: [0..1]

Definition: A country with its own government

Data type: CountryCode

Data format: [A-Z]{2,2}

Rule: Country code is checked against the list of country names under the ISO 3166 (2 alphanumeric characters code version).

JSON example of element:

```
"country": "CZ"
```

4.34.2.6 addressLine (Unstructured address record)

JSON record: ultimateDebtor.postalAddress.addressLine

Occurrence only SEPA payments: [0..2]

Definition: Information that localizes and identifies the specific address as defined by postal services, presented in the free text format.

Data type: Max70Text

Maximum 2 lines with 70 characters per line

4.34.3 identification (Identification)

JSON record: *ultimateDebtor*. identification

Occurrence: only SEPA payments: [0..1]

Definition: A unique and unambiguous identification of the party.

Type: This message element consists of the following elements **Party6Choice**:

LEVEL	OR	MESSAGE ELEMENT	OCCURRENCE	PRESENTATION
+++	{Or	organisationIdentification	[1..1]	Identification of organization
+++	Or}	privateIdentification	[1..1]	Identification of private individual

JSON example of element:

```
"identification": {
  "privateIdentification": {
    "other": [
      {
        "identification": "57842578842",
        "schemeName": {
          "proprietary": "passport",
          "issuer": "WIENA"
        }
      }
    ]
  }
}
```

4.35 valueDate (Due date)

JSON record: *valueDate.date*, *bookingDate.date*

Occurrence: [0..1]

Definition: Due date/payment currency

Application: Depending on the Due date / payment currency and the way the bank presents data (and time)

Type: This message element consists of the following elements **Party6Choice:**

LEVEL	MESSAGE ELEMENT	OCCURRENCE	PRESENTATION
+++	date	[1..1]	Date of processing

JSON example of element:

```
"valueDate": {  
  "date": "2016-09-05T00:00:00+01:00"  
}
```

4.35.1 date (Date)

JSON record: valueDate.date, bookingDate.date

Definition: Date of processing/posting the payment by the bank.

Application: Format ISODate i.e. YYYY-MM-DD, or ISODateTime i.e. YYYY-MM-DDThh:mm:ss.sTZD is depending on the transaction type and the way the bank presents data (and time) of payment processing/posting where:

YYYY = four-digit year

MM = two-digit month (01=January, etc.)

DD = two-digit day of month (01 through 31)

hh = two digits of hour (00 through 23) (am/pm NOT allowed)

mm = two digits of minute (00 through 59)

ss = two digits of second (00 through 59)

TZD = time zone designator (Z or +hh:mm or -hh:mm)"

Format type: ISODate or ISODateTime

JSON example of element:

```
"date": "2016-09-05T00:00:00+01:00"
```

5 Examples of Request and Response for Individual Messages

5.1 Example of request and response for Query for Balance Check at account.

5.1.1 Example of request

Request headers:

```
POST https://api.bank.cz/v1/accounts/balanceCheck
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:20:01 GMT
User-Involved: false
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
```

Request body:

```
{
  "exchangeIdentification": "658576010faf0a23dc",
  "card": {
    "cardHolderName": "Jan Novák",
    "maskedPAN": "1234*****6789"
  },
  "debtorAccount": {
    "identification": {
      "iban": "CZ0708000000001019382023"
    },
    "currency": "CZK"
  },
  "authenticationMethod": "NPIN",
  "merchant": {
    "identification": "471 16 129",
```



```
"shortName": "NEOLUXOR",
"commonName": "Neoluxor s.r.o.",
"address": "Hlavní 5, Praha 1",
"countryCode": "CZ",
"merchantCategoryCode": "5192"
},
"transactionDetails": {
  "currency": "CZK",
  "totalAmount": 10050.15
}
}
```

5.1.2 Example of correct response

Response **200 OK** Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response **200 OK** Body

```
{
  "responseIdentification": 98765,
  "exchangeIdentification": "658576010faf0a23dc",
  "response": "APPR"
}
```

5.1.3 Example of response to wrong call

Response **400 Bad Request** Headers

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 400 Bad Request Body

```
{
  "errors": [
    {
      "error": "FIELD_MISSING",
      "scope": "merchant.identification",
      "parameters": null,
      "message": null
    },
    {
      "error": "AC09",
      "scope": null,
      "parameters": null,
      "message": null
    }
  ]
}
```

5.2 Example of request and response for Query for list of accounts of bank's client

5.2.1 Example of request

Request headers:

```
GET https://api.bank.cz/v1/my/accounts
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:21:01 GMT
User-Involved: true
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456
```

5.2.2 Example of correct response

Response 200 OK Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 200 OK Body

```
{
  "pageNumber": 0,
  "pageCount": 2,
  "pageSize": 100,
  "nextPage": 1,
  "accounts": [
    {
      "id": "D2C8C1DCC51A3738538A40A4863CA288E0225E52",
      "identification": {
        "iban": "CZ0708000000001019382023",
        "other": "1019382023"
      },
      "currency": "CZK",
      "servicer": {
        "bankCode": "0800",
        "countryCode": "CZ",
        "bic": "GIBACZPX"
      },
      "nameI18N": "Muj hlavni osobni ucet",
      "productI18N": "Osobní účet ČS",
      "ownersNames": [
        "Jan Novák"
      ],
      "relationship": {
        "isOwner": true
      }
    }
  ]
}
```

```

    },
    "suitableScope": {
        "AISP": "",
        "PISP": " paymentOrders paymentOrders.instantPayment
standingOrders",
        "CISP": ""
    }
}
]
}

```

5.2.3 Example of response to wrong call

Response 400 Bad Request Headers

```

HTTP/1.1 400 Bad Request
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4

```

Response 400 Bad Request Body

```

{
  "errors": [
    {
      "error": "PAGE_NOT_FOUND"
    },
    {
      "error": "PARAMETER_INVALID",
      "scope": "sort"
    }
  ]
}

```

5.3 Example of request and response for Query for balance in a specific account of bank's client

5.3.1 Example of request

Request headers:

```
GET
https://api.bank.cz/v1/my/accounts/D2C8C1DCC51A3738538A40A4863CA288E02
25E52/balance

Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:22:01 GMT
User-Involved: true
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456
```

5.3.2 Example of correct response

Response 200 OK Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 200 OK Body

```
{
  "balances": [
    {
      "type": {
        "codeOrProprietary": {
          "code": "PRCD"
        }
      },
      "creditLine": {
        "included": true,

```

```

        "amount": {
            "value": 10000.00,
            "currency": "CZK"
        }
    },
    "amount": {
        "value": 4520.15,
        "currency": "CZK"
    },
    "creditDebitIndicator": "DBIT",
    "date": {
        "dateTime": "2017-02-17T12:32:41.0Z"
    }
}
]
}

```

5.3.3 Example of response to wrong call

5.3.3.1 Error 404

Response **404 Not Found** Headers

```

HTTP/1.1 404 Not Found
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4

```

Response **404 Not Found** Body

```

{
  "errors": [
    {
      "error": "ID_NOT_FOUND"
    }
  ]
}

```

```
}  
]  
}
```

5.3.3.2 Error 400

Response **400 Bad Request** Headers

```
HTTP/1.1 400 Bad Request  
Content-Type: application/json  
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response **400 Bad Request** Body

```
{  
  "errors": [  
    {  
      "error": "AM03",  
      "scope": "currency"  
    }  
  ]  
}
```

5.4 Example of request and response for Query for a transaction overview in a specific account of bank's client

5.4.1 Example of request

Request headers:

```
GET  
https://api.bank.cz/v1/my/accounts/D2C8C1DCC51A3738538A40A4863CA288E02  
25E52/transactions  
Content-Type: application/json  
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4  
Date: Wed, 6 Jan 2019 07:23:01 GMT  
User-Involved: false
```

```
API-key: 00000000-1212-0f0f-a0a0-123456789abc
```

```
TPP-Name: Google China
```

```
Authorization: Bearer AbCdEf123456
```

5.4.2 Example of correct response

Response 200 OK Headers

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
```

```
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 200 OK Body

```
{
  "pageNumber": 0,
  "pageCount": 2,
  "pageSize": 100,
  "nextPage": 1,
  "transactions": [
    {
      "entryReference": "RB-4567813",
      "amount": {
        "value": 10000.00,
        "currency": "CZK"
      },
      "status": "BOOK",
      "creditDebitIndicator": "DBIT",
      "bookingDate": {
        "date": "2017-01-31T00:00:00.000+01"
      },
      "valueDate": {
        "date": "2017-01-31T00:00:00.000+01"
      },
      "bankTransactionCode": {
```



```

    "proprietary": {
      "code": 000001000010,
      "issuer": "CBA"
    }
  },
  "entryDetails": {
    "transactionDetails": {
      "amountDetails": {
        "instructedAmount": {
          "amount": {
            "value": 10000.00,
            "currency": "CZK"
          }
        }
      }
    },
    "relatedParties": {
      "debtor": {
        "name": "Novák Jan"
      },
      "debtorAccount": {
        "identification": {
          "iban": "CZ0827000000002108589434",
          "other": {
            "identification": "0000002108589434"
          }
        }
      }
    }
  },
  "relatedAgents": {
    "debtorAgent": {
      "financialInstitutionIdentification": {
        "bic": "BACXCZPP",

```

```

        "clearingSystemMemberIdentification": {
            "memberIdentification": "2700"
        }
    },
    "remittanceInformation": {
        "unstructured": "",
        "structured": {
            "creditorReferenceInformation": {
                "reference":
"VS:123456\","KS:456789\","SS:879213546"
            }
        }
    },
    "additionalTransactionInformation": "Domáci platba -
S24/IB,záloha plyn Bohemia Energy",
    "additionalRemittanceInformation": "Platba za plyn -
Bohemia Energy"
}
},
{
    "amount": {
        "value": 105.25,
        "currency": "CZK"
    },
    "status": "BOOK",
    "creditDebitIndicator": "DBIT",
    "bookingDate": {
        "date": "2016-09-05T00:00:00+01:00"
    },
    "valueDate": {

```

```

        "date": "2016-09-05T00:00:00+01:00"
    },
    "bankTransactionCode": {
        "proprietary": {
            "code": 4000050,
            "issuer": "CBA"
        }
    },
    "entryDetails": {
        "transactionDetails": {
            "references": {
                "chequeNumber": "xxxxxxxxxxxx1248"
            },
            "amountDetails": {
                "instructedAmount": {
                    "amount": {
                        "value": 10.00,
                        "currency": "GBP"
                    }
                }
            },
            "counterValueAmount": {
                "amount": {
                    "currency": "CZK",
                    "value": 105.25
                }
            },
            "currencyExchange": {
                "sourceCurrency": "CZK",
                "targetCurrency": "GBP",
                "exchangeRate": 10.525
            }
        }
    },
    },

```

```

        "additionalTransactionInformation": "PLATBA KARTOU"
    }
}
},
{
    "entryReference": "FC-4567513951",
    "amount": {
        "value": 1844777.00,
        "currency": "CZK"
    },
    "status": "BOOK",
    "creditDebitIndicator": "CRDT",
    "bookingDate": {
        "date": "2017-01-31T00:00:00.000+01"
    },
    "valueDate": {
        "date": "2017-01-31T00:00:00.000+01"
    },
    "bankTransactionCode": {
        "proprietary": {
            "code": 00001000020,
            "issuer": "CBA"
        }
    },
    "entryDetails": {}
},
{
    "entryReference": "CDR-13457893331",
    "amount": {
        "value": 2.00,
        "currency": "CZK"
    },
}

```

```

    "status": "BOOK",
    "creditDebitIndicator": "DBIT",
    "bookingDate": {
      "date": "2016-09-05T00:00:00+01:00"
    },
    "valueDate": {
      "date": "2016-09-05T00:00:00+01:00"
    },
    "bankTransactionCode": {
      "proprietary": {
        "code": 00004000010,
        "issuer": "CBA"
      }
    },
    "entryDetails": {
      "transactionDetails": {
        "amountDetails": {
          "instructedAmount": {
            "amount": {
              "value": 2.00,
              "currency": "CZK"
            }
          }
        }
      },
      "additionalTransactionInformation": "POPLATEK ZA ODCHOZÍ
TRANSAKČÍ"
    }
  },
  {
    "amount": {
      "value": 122.22,
      "currency": "CZK"
    }
  }
}

```

```

    },
    "status": "BOOK",
    "creditDebitIndicator": "CRDT",
    "bookingDate": {
      "date": "2016-09-05T00:00:00+01:00"
    },
    "valueDate": {
      "date": "2016-09-05T00:00:00+01:00"
    },
    "bankTransactionCode": {
      "proprietary": {
        "code": 00009000020,
        "issuer": "CBA"
      }
    },
    "entryDetails": {
      "transactionDetails": {
        "amountDetails": {
          "instructedAmount": {
            "amount": {
              "value": 122.22,
              "currency": "CZK"
            }
          }
        }
      },
      "additionalTransactionInformation": "PŘIPSÁNÍ ÚROKU ZE
ZUSTATKU"
    }
  },
  {
    "entryReference": "FP-4156489123",
    "amount": {

```

```

        "value": 23282.62,
        "currency": "CZK"
    },
    "status": "BOOK",
    "creditDebitIndicator": "CRDT",
    "bookingDate": {
        "date": "2017-01-31T00:00:00.000+01"
    },
    "valueDate": {
        "date": "2017-01-31T00:00:00.000+01"
    },
    "bankTransactionCode": {
        "proprietary": {
            "code": 00001000040,
            "issuer": "CBA"
        }
    },
    "entryDetails": {
        "transactionDetails": {
            "references": {
                "endToEndIdentification":
"vs0250117002/SS00000000000/KS0000"
            },
            "amountDetails": {
                "instructedAmount": {
                    "amount": {
                        "value": 23282.62,
                        "currency": "CZK"
                    }
                }
            },
            "counterValueAmount": {
                "amount": {
                    "currency": "EUR",

```

```

        "value": 86200.00
    },
    "currencyExchange": {
        "sourceCurrency": "EUR",
        "targetCurrency": "CZK",
        "exchangeRate": 27.01
    }
},
"relatedParties": {
    "debtor": {
        "name": "RENWORTH s.r.o",
        "identification": {
            "organisationIdentification": {
                "other": [
                    {
                        "identification": "48135283",
                        "schemeName": {
                            "proprietary": "partnership agreement"
                        }
                    }
                ]
            }
        }
    },
    "debtorAccount": {
        "identification": {
            "iban": "CZ1308001800640033122856"
        }
    }
},
"relatedAgents": {

```



```

        "debtorAgent": {
            "financialInstitutionIdentification": {
                "bic": "GIBACZPXXXX"
            }
        },
        "purpose": {
            "proprietary": "PLATBA ZA SLUŽBY"
        },
        "remittanceInformation": {
            "structured": {
                "creditorReferenceInformation": {
                    "reference": "VS:0250117002"
                }
            }
        },
        "additionalTransactionInformation": "8201701069595 BIC:
GIBACZPXXXX; #71A# SHA ZALOHA DLE SMLOUVY O DODAVKACH,zaloha dle
smlouvy o dodavkach c. 45678/2017,VS0250117002/SS0000000000/KS0000SEPA
převod"
    },
    {
        "amount": {
            "value": 105.00,
            "currency": "CZK"
        },
        "status": "BOOK",
        "creditDebitIndicator": "CRDT",
        "bookingDate": {
            "date": "2016-09-05T00:00:00+01:00"
        }
    },

```

```
    "valueDate": {
      "date": "2016-09-05T00:00:00+01:00"
    },
    "bankTransactionCode": {
      "proprietary": {
        "code": 00002000010,
        "issuer": "CBA"
      }
    }
  }
]
```

5.4.3 Example of response to wrong call

5.4.3.1 Error 404

Response **404 Not Found** Headers

```
HTTP/1.1 404 Not Found
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response **404 Not Found** Body

```
{
  "errors": [
    {
      "error": "ID_NOT_FOUND"
    }
  ]
}
```

5.4.3.2 Error 400

Response 400 Bad Request Headers

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 400 Bad Request Body

```
{
  "errors": [
    {
      "error": "AM03",
      "scope": "currency"
    },
    {
      "error": "DT01",
      "parameters": {
        "DATE": "DATE_TO_OLD"
      },
      "scope": "fromDate"
    },
    {
      "error": "DT01",
      "parameters": {
        "DATE": "DATE_IN_FUTURE"
      },
      "scope": "toDate"
    }
  ]
}
```

5.5 Example of request and response for Query for a awaiting transaction overview in a specific account of bank's client

5.5.1 Example of request

Request headers:

```
GET
https://api.bank.cz/v1/my/accounts/D2C8C1DCC51A3738538A40A4863CA288E02
25E52/payments/awaiting

Content-Type: application/json

X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa9874d5ac2

Date: Wed, 1 Jan 2020 08:13:01 GMT

User-Involved: false

API-key: 00000000-1212-0f0f-a0a0-123456789abc

TPP-Name: Google China

Authorization: Bearer AbCdEf123456
```

5.5.2 Example of correct response

Response 200 OK Headers

```
HTTP/1.1 200 OK

Content-Type: application/json

X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa9874d5ac2
```

Response 200 OK Body

```
{
  "pageNumber": 0,
  "pageCount": 1,
  "nextPage": 0,
  "pageSize": 10,
  "totalCount": 1,
  "payments": [
    {
      "paymentIdentification": {
        "instructionIdentification": "None",
```

```

    "transactionIdentification": "CP151255555"
  },
  "paymentTypeInformation": {
    "serviceLevel": {
      "code": "DMCT"
    }
  },
  "amount": {
    "instructedAmount": {
      "value": 3000,
      "currency": "CZK"
    }
  },
  "requestedExecutionDate": "2019-12-22",
  "debtorAccount": {
    "id": "CUA1234567899",
    "identification": {
      "iban": "CZ8858000000001234567899",
      "other": {
        "identification": "1234567899"
      }
    }
  },
  "creditorAgent": {
    "financialInstitutionIdentification": {
      "clearingSystemMemberIdentification": {
        "clearingSystemIdentification": {
          "code": "0100"
        }
      }
    }
  },

```

```

    "creditorAccount": {
      "identification": {
        "other": {
          "identification": "000019-2784000277"
        }
      }
    },
    "remittanceInformation": {
      "structured": {
        "creditorReferenceInformation": {
          "reference": [
            "VS:20200312"
          ]
        }
      }
    },
    "awaitingStatus": "OVERDUE"
  }
]
}

```

5.5.3 Example of response to wrong call

5.5.3.1 Error 404

Response **404 Not Found** Headers

```

HTTP/1.1 404 Not Found
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4

```

Response **404 Not Found** Body

```

{
  "errors": [
    {

```

```
    "error": "ID_NOT_FOUND"
  }
]
}
```

5.5.3.2 Error 400

Response 400 Bad Request Headers

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 400 Bad Request Body

```
{
  "errors": [
    {
      "error": "AM03",
      "scope": "currency"
    },
    {
      "error": "DT01",
      "parameters": {
        "DATE": "DATE_TO_OLD"
      },
      "scope": "fromDate"
    },
    {
      "error": "DT01",
      "parameters": {
        "DATE": "DATE_IN_FUTURE"
      },
      "scope": "toDate"
    }
  ]
}
```

```
}  
]  
}
```

5.6 Example of request and response after entering a new payment

5.6.1 Example of request Domestic payment

Request headers:

```
POST https://api.bank.cz/v1/my/payments  
Content-Type: application/json  
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4  
Date: Wed, 6 Jan 2019 07:24:01 GMT  
User-Involved: true  
API-key: 00000000-1212-0f0f-a0a0-123456789abc  
TPP-Name: Google China  
Authorization: Bearer AbCdEf123456
```

Request Body:

```
{  
  "paymentIdentification": {  
    "instructionIdentification": "NejakeID41785962314574"  
  },  
  "paymentTypeInformation": {  
    "instructionPriority": "NORM"  
  },  
  "amount": {  
    "instructedAmount": {  
      "value": 1245.44,  
      "currency": "CZK"  
    }  
  },  
}
```



```

    "requestedExecutionDate": "2017-01-31",
    "debtorAccount": {
      "identification": {
        "iban": "CZ7508000000002108589434"
      },
      "currency": "CZK"
    },
    "creditorAccount": {
      "identification": {
        "iban": "CZ633030000000000000123"
      },
      "currency": "CZK"
    },
    "remittanceInformation": {
      "unstructured": "/VS/7418529630/SS/1234567890",
      "additionalRemittanceInformation":
        "Platba za MŠ ..."
    }
  }
}

```

5.6.2 Example of request SEPA payment

Request headers:

```

POST https://api.bank.cz/v1/my/payments
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:25:01 GMT
User-Involved: true
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456

```

Request Body:

```

{
  "paymentIdentification": {
    "instructionIdentification": "SEPAFUL123",
    "endToEndIdentification": "NOTPROVIDED"
  },
  "paymentTypeInformation": {
    "instructionPriority": "HIGH"
  },
  "amount": {
    "instructedAmount": {
      "value": 45789.45,
      "currency": "EUR"
    }
  },
  "requestedExecutionDate": "2017-01-31",
  "ultimateDebtor": {
    "name": "Karel Novák",
    "postalAddress": {
      "streetName": "Vodičkova",
      "buildingNumber": "12",
      "postCode": "12000",
      "townName": "Praha 2",
      "country": "CZ"
    }
  },
  "identification": {
    "privateIdentification": {
      "other": [
        {
          "identification": "57842578842",
          "schemeName": {
            "proprietary": "passport",
            "issuer": "WIENA"
          }
        }
      ]
    }
  }
}

```

```

    }
  }
]
}
},
  "debtorAccount": {
    "identification": {
      "iban": "CZ7508000000002108589434"
    },
    "currency": "CZK"
  },
  "creditorAgent": {
    "financialInstitutionIdentification": {
      "bic": "GIBAATWWXXX"
    }
  },
  "creditor": {
    "name": "1. wiena investment",
    "postalAddress": {
      "streetName": "Reisnerstraße",
      "buildingNumber": "20",
      "postCode": "1030",
      "townName": "Wiena",
      "country": "AT"
    }
  },
  "creditorAccount": {
    "identification": {
      "iban": "AT872011102000123456"
    }
  },
},

```

```

"ultimateCreditor": {
  "name": "Franz Schubert",
  "postalAddress": {
    "streetName": "Dianagasse",
    "buildingNumber": "6",
    "postCode": "1030",
    "townName": "Wiena",
    "country": "AT"
  },
  "identification": {
    "privateIdentification": {
      "other": [
        {
          "identification": "57842578842",
          "schemeName": {
            "proprietary": "passport",
            "issuer": "WIENA"
          }
        }
      ]
    }
  },
  "purpose": {
    "proprietary": "sale of shares"
  }
}

```

5.6.3 Example of request Foreign payment within EEA

Request headers:

```

POST https://api.bank.cz/v1/my/payments
Content-Type: application/json

```

```
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:26:01 GMT
User-Involved: true
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456
```

Request Body:

```
{
  "paymentIdentification": {
    "instructionIdentification": "MOJeID1234"
  },
  "paymentTypeInformation": {
    "instructionPriority": "NORM"
  },
  "amount": {
    "instructedAmount": {
      "value": 1245.44,
      "currency": "USD"
    }
  },
  "requestedExecutionDate": "2017-01-31",
  "debtorAccount": {
    "identification": {
      "iban": "CZ8601000000000161486937"
    },
    "currency": "CZK"
  },
  "creditorAgent": {
    "financialInstitutionIdentification": {
```

```

        "bic": "GIBACZPX"
    }
},
    "creditor": {
        "name": "Bohumil Hrabal",
        "postalAddress": {
            "streetName": "Na Hrázi",
            "buildingNumber": "326/24",
            "postCode": "18000",
            "townName": "Praha 8",
            "country": "CZ"
        }
    },
    "creditorAccount": {
        "identification": {
            "iban": "CZ3908000000000204533335",
            "other": {
                "identification": "204533335"
            }
        }
    },
    "remittanceInformation": {
        "unstructured": "fa 123546897"
    }
}

```

5.6.4 Example of request Foreign payment outside EEA

Request headers:

```

POST https://api.bank.cz/v1/my/payments
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:27:01 GMT

```

```
User-Involved: true
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456
```

Request Body:

```
{
  "paymentIdentification": {
    "instructionIdentification": "IDTPP45678911"
  },
  "paymentTypeInformation": {
    "instructionPriority": "NORM"
  },
  "amount": {
    "instructedAmount": {
      "value": 1245.44,
      "currency": "GBP"
    }
  },
  "requestedExecutionDate": "2017-01-31",
  "debtorAccount": {
    "identification": {
      "iban": "CZ6330300000000000000123"
    },
    "currency": "CZK"
  },
  "creditorAgent": {
    "financialInstitutionIdentification": {
      "bic": "ABNYUS33",
      "name": "New York Commercial Bank",
      "postalAddress": {
        "streetName": "Merrick Avenue615",
```

```

        "buildingNumber": "61511590-6644",
        "postCode": "11590-6644",
        "townName": "WESTBURY, NY",
        "country": "US"
    }
}
},
"creditor": {
    "name": "First Hudson boat Inc.",
    "postalAddress": {
        "streetName": "Grand St",
        "buildingNumber": "1256",
        "postCode": "NY 11211",
        "townName": "Brooklyn ,NY",
        "country": "US"
    }
},
"creditorAccount": {
    "identification": {
        "other": {
            "identification": "123456789"
        }
    }
}
}
}
}

```

5.6.5 Example of correct response

Response 200 OK Headers

```

HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4

```


Response 200 OK Body

```
{
  "paymentIdentification": {
    "instructionIdentification": "NOTPROVIDED",
    "transactionIdentification": "048885570000001020045"
  },
  "paymentTypeInformation": {
    "instructionPriority": "NORM",
    "serviceLevel": {
      "code": "DMCT"
    }
  },
  "amount": {
    "instructedAmount": {
      "value": 10050.15,
      "currency": "CZK"
    }
  },
  "requestedExecutionDate": "2017-02-20",
  "debtorAccount": {
    "identification": {
      "iban": "CZ0708000000001019382023"
    },
    "currency": "CZK"
  },
  "creditorAccount": {
    "identification": {
      "iban": "CZ0708000000001019540081"
    },
    "currency": "CZK"
  },
  "remittanceInformation": {
```

```

    "structured": {
      "creditorReferenceInformation": {
        "reference": [
          "VS:501",
          "KS:9",
          "SS:1005"
        ]
      }
    },
    "signInfo": {
      "state": "OPEN",
      "signId": "164298331754922"
    },
    "instructionStatus": "ACWC",
    "statusChangeInfo": "DT01"
  }

```

5.6.6 Example of responses to wrong call

Response 400 Bad Request Headers

```

HTTP/1.1 400 Bad Request
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4

```

Response 400 Bad Request Body

```

{
  "errors": [
    {
      "error": "AC01"
    },
    {
      "error": "AM06",

```

```

    "scope": "amount.instructedAmount.value"
  },
  {
    "error": "AM03",
    "parameters": {
      "CURRENCY": "JPY",
      "CURRENCY_ALLOWED": "CZK"
    },
    "scope": "debtorAccount.currency"
  }
]
}

```

Response **400 Bad Request** on request for initiating an instant payment where the counterparty bank is not involved in the scheme

```

{
  "errors": [
    {
      "error": "AM03",
      "parameters": {
        "ALLOWED_PRIORITY_TYPES": ["NORM", "HIGH"]
      },
      "scope": "debtorAccount.currency"
    }
  ]
}

```

5.7 Example of request and response for Query for status of entered/initiated payment

5.7.1 Example of request

Request headers:

```
GET https://api.bank.cz/v1/my/payments/048885570000001020045/status
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:28:01 GMT
User-Involved: false
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456
```

5.7.2 Example of correct response

Response **200 OK** Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response **200 OK** Body:

```
{
  "instructionStatus": "ACTC"
}
```

5.7.3 Example of response to wrong call

Response **404 Not Found** Headers

```
HTTP/1.1 404 Not Found
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response **404 Not Found** Body

```
{
  "errors": [
    {
      "error": "TRANSACTION_MISSING"
    }
  ]
}
```

```
]
}
```

5.8 Example of request and response for Query for info on entered/initiated payment

5.8.1 Example of request

Request headers:

```
GET https://api.bank.cz/v1/my/payments/048885570000001020045
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:29:01 GMT
User-Involved: true
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456
```

5.8.2 Example of correct response

Response 200 OK Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 200 OK Body:

```
{
  "paymentIdentification": {
    "instructionIdentification": "NOTPROVIDED",
    "transactionIdentification": "048885570000001020045"
  },
  "paymentTypeInformation": {
    "instructionPriority": "NORM",
    "serviceLevel": {
```

```

        "code": "DMCT"
    }
},
    "amount": {
        "instructedAmount": {
            "value": 10050.15,
            "currency": "CZK"
        }
    },
    "requestedExecutionDate": "2017-02-20",
    "debtorAccount": {
        "identification": {
            "iban": "CZ0708000000001019382023"
        }
    },
    "currency": "CZK"
},
    "creditorAccount": {
        "identification": {
            "iban": "CZ0708000000001019540081"
        }
    },
    "currency": "CZK"
},
    "remittanceInformation": {
        "structured": {
            "creditorReferenceInformation": {
                "reference": [
                    "VS:501",
                    "KS:9",
                    "SS:1005"
                ]
            }
        }
    }
}

```

```

    },
    "signInfo": {
      "state": "OPEN",
      "signId": "164298331754922"
    },
    "instructionStatus": "ACWC",
    "statusChangeInfo": "TM01"
  }
}

```

5.8.3 Example of response to wrong call

Response 404 Not Found Headers

```

HTTP/1.1 404 Not Found
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4

```

Response 404 Not Found Body

```

{
  "errors": [
    {
      "error": "TRANSACTION_MISSING"
    }
  ]
}

```

5.9 Example of request and response to delete unauthorised and authorised payments

5.9.1 Example of request

Request headers:

```

DELETE https://api.bank.cz/v1/my/payments/048885570000001020045
Content-Type: application/json

```

```
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:30:01 GMT
User-Involved: true
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456
```

5.9.2 Example of correct response

Response **204 No Content** Headers – in case of unauthorised payment

```
HTTP/1.1 204 No Content
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response **200 OK** Headers – in case of authorised payment

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response **200 OK** Body – in case of authorised payment

```
{
  "paymentIdentification": {
    "instructionIdentification": "SomeID-1285673519924",
    "endToEndIdentification": "abcd557000000102",
    "transactionIdentification": "048885570000001020045"
  },
  "signInfo": {
    "state": "OPEN",
    "signId": "164298331754922"
  }
}
```

5.9.3 Example of response to wrong call

Response **404 Not Found** Headers


```
HTTP/1.1 404 Not Found
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 404 Not Found Body

```
{
  "errors": [
    {
      "error": "TRANSACTION_MISSING"
    }
  ]
}
```

5.10 Example of request and response to generate payment authorization ID

5.10.1 Example of request

Request headers:

```
POST https://api.bank.cz/v1/my/payments/048885570000001020045/sign
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:32:01 GMT
User-Involved: false
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456
```

5.10.2 Example of correct response

Response 200 OK Headers

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response **200 OK** Body:

```
{
  "scenarios": [
    "CODE of The One of the methods ..."
    "CODE of The second method ..."
  ],
  "signInfo": {
    "state": "OPEN",
    "signId": "164298331754922"
  }
}
```

5.10.3 Example of response to wrong call

Response **404 Not Found** Headers

```
HTTP/1.1 404 Not Found
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response **404 Not Found** Body

```
{
  "errors": [
    {
      "error": "TRANSACTION_MISSING"
    }
  ]
}
```

5.11 Example of request and response for Payment Authorization - Step I. Payment Authorization Detail

5.11.1 Example of request

Request headers:

```
GET
https://api.bank.cz/v1/my/payments/048885570000001020045/sign/164298331754922

Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:33:01 GMT
User-Involved: true
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456
```

5.11.2 Example of correct response

Response 200 OK Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 200 OK Body:

```
{
  "scenarios": [
    "CODE of The One of the methods ..."
    "CODE of The second method ..."
  ],
  "signInfo": {
    "state": "OPEN",
    "signId": "164298331754922"
  }
}
```

5.11.3 Example of response to wrong call

5.11.3.1 Error 404

Response **404 Not Found** Headers

```
HTTP/1.1 404 Not Found
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response **404 Not Found** Body

```
{
  "errors": [
    {
      "error": "ID_NOT_FOUND",
      "scope": "signId"
    },
    {
      "error": "ID_NOT_FOUND",
      "scope": "paymentId"
    }
  ]
}
```

5.11.3.2 Error 400

Response **400 Bad Request** Headers

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response **400 Bad Request** Body

```
{
```

```
"errors": [  
  {  
    "error": "AUTH_LIMIT_EXCEEDED"  
  }  
]  
}
```

5.12 Example of request and response for Payment Authorization - Step II. Payment Authorization Initiation - specific for each bank

5.12.1 Example of requests

Request headers:

```
POST  
https://api.bank.cz/v1/my/payments/048885570000001020045/sign/16429833  
1754922  
Content-Type: application/json  
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4  
Date: Wed, 6 Jan 2019 07:34:01 GMT  
User-Involved: true  
API-key: 00000000-1212-0f0f-a0a0-123456789abc  
TPP-Name: Google China  
Authorization: Bearer AbCdEf123456
```

Request Body only with the selected method:

```
{  
  "authorizationType": "SMS"  
}
```

Request Body only with the selected method:

```
{  
  "authorizationType": "SMS",  
}
```

```
"redirectUrl": "http://my.sometpp.com/..."
}
```

5.12.2 Example of correct response

Response **200 OK** Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response **200 OK** Body:

```
{
  "authorizationType": "USERAGENT_REDIRECT",
  "href": {
    "url": "http://www.bank.cz/authorization/164298331754922",
    "id": "164298331754922"
  },
  "method": "GET",
  "formData": {
    "SAMLRequest": "45sa4d4e4e4fds5f6s4df4sd6f",
    "relayState": "token"
  },
  "signInfo": {
    "state": "OPEN",
    "signId": "164298331754922"
  }
}
```

5.12.3 Example of response to wrong call

5.12.3.1 Error 404

Response **404 Not Found** Headers

```
HTTP/1.1 404 Not Found
```

```
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 404 Not Found Body

```
{
  "errors": [
    {
      "error": "ID_NOT_FOUND"
    }
  ]
}
```

5.12.3.2 Error 400

Response 400 Bad Request Headers

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 400 Bad Request Body

```
{
  "errors": [
    {
      "error": "AUTH_LIMIT_EXCEEDED"
    }
  ]
}
```

5.13 Example of request and response for Payment Authorization - Step III. Payment Authorization Finalization - specific for each bank

5.13.1 Example of request

Request headers:

```
PUT
https://api.bank.cz/v1/my/payments/048885570000001020045/sign/16429833
1754922

Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:35:01 GMT
User-Involved: true
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456
```

Request Body:

```
{
  "authorizationType": "SMS",
  "oneTimePassword": "12345"
}
```

5.13.2 Example of correct response

Response 200 OK Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 200 OK Body:

```
{
  "state": "DONE",
  "instructionStatus": "ACTC",
  "pollInterval": 5000
}
```


5.13.3 Example of response to wrong call

Response 404 Not Found Headers

```
HTTP/1.1 404 Not Found
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 404 Not Found Body

```
{
  "errors": [
    {
      "error": "ID_NOT_FOUND"
    }
  ]
}
```

5.14 Example of request and response for list of client's standing orders

5.14.1 Example of request

Request headers:

```
GET https://api.bank.cz/v1/my/standingorders?size=25&page=0
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:36:01 GMT
User-Involved: false
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456
```

5.14.2 Example of correct response

Response 200 OK Headers

```
HTTP/1.1 200 OK
```

Content-Type: application/json

X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4

Response **200 OK** Body:

```
{
  "pageNumber": 0,
  "pageCount": 1,
  "pageSize": 25,
  "nextPage": 0,
  "standingOrders": [{
    "standingOrderIdentification": {
      "instructionIdentification": "SomeID-1285673519924",
      "transactionIdentification": "033225570000078020091"
    },
    "amount": {
      "instructedAmount": {
        "value": 1500.00,
        "currency": "CZK"
      }
    },
    "standingOrder": {
      "alias": "Payments for lunch - kindergarten Michal",
      "execution": {
        "interval": "MONTHLY",
        "intervalDue": "25",
        "mode": "MAX_AMOUNT_EXCEEDED",
        "modeDue": "DUE_OR_BEFORE_DAY_OF_MONTH"
      }
    },
    "debtorAccount": {
      "id": "0f45a992....",
      "identification": {
```

```
        "iban": "CZ7508000000002108589434"
      },
      "currency": "CZK"
    }
  ]
}
```

5.14.3 Example of response to wrong call

Response 400 **Bad Response Headers**

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 400 **Bad Response Body**

```
{
  "errors": [
    {
      "error": "PAGE_NOT_FOUND"
    }
  ]
}
```

5.15 Example of request and response after entering a new standing order

5.15.1 Example of request standing order initiation for Domestic payment

Request headers:

```
POST https://api.bank.cz/v1/my/standingorders
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

```
Date: Wed, 6 Jan 2019 07:38:01 GMT
User-Involved: true
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456
```

Request Body:

```
{
  "standingOrderIdentification": {
    "instructionIdentification": "SomeID-1285673519924"
  },
  "amount": {
    "instructedAmount": {
      "value": 1500.00,
      "currency": "CZK"
    }
  },
  "requestedExecutionDate": "2018-09-08",
  "standingOrder": {
    "alias": "Payments for lunch - kindergarten Michal",
    "execution": {
      "interval": "MONTHLY",
      "intervalDue": "25",
      "mode": "MAX_AMOUNT_EXCEEDED",
      "modeDue": "DUE_OR_BEFORE_DAY_OF_MONTH"
    },
    "exceptions": {
      "stoppages": ["JULY", "AUGUST"],
      "breaks": [
        {
          "validFromDate": "2018-11-01",
          "validToDate": "2019-01-30"
        }
      ]
    }
  }
}
```

```

    },
    {
      "validFromDate": "2019-05-01",
      "validToDate": "2019-06-15"
    }
  ]
},
"validity": {
  "lastExecutionDate": null,
  "maxAmount": {
    "value": 89000.00,
    "currency": "CZK"
  }
}
},
"debtorAccount": {
  "identification": {
    "iban": "CZ7508000000002108589434"
  },
  "currency": "CZK"
},
"creditorAccount": {
  "identification": {
    "iban": "CZ6330300000000000000123"
  },
  "currency": "CZK"
},
"remittanceInformation": {
  "unstructured": "/VS/7418529630/SS/1234567890"
}
}

```

5.15.2 Example of correct response

Response 200 OK Headers:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 200 OK Body

```
{
  "standingOrderIdentification": {
    "instructionIdentification": "SomeID-1285673519924",
    "transactionIdentification": "033225570000078020091"
  },
  "paymentTypeInformation": {
    "serviceLevel": {
      "code": "DMCT"
    }
  },
  "amount": {
    "instructedAmount": {
      "value": 1500.00,
      "currency": "CZK"
    }
  },
  "requestedExecutionDate": "2018-09-08",
  "standingOrder": {
    "alias": "Payments for lunch - kindergarten Michal",
    "execution": {
      "interval": "MONTHLY",
      "intervalDue": "25",
      "mode": "MAX_AMOUNT_EXCEEDED",
      "modeDue": "DUE_OR_BEFORE_DAY_OF_MONTH"
    },
    "exceptions": {
```

```

    "stoppages": ["JULY", "AUGUST"],
    "breaks": [
      {
        "validFromDate": "2018-11-01",
        "validToDate": "2019-01-30"
      },
      {
        "validFromDate": "2019-05-01",
        "validToDate": "2019-06-15"
      }
    ],
  },
  "validity": {
    "lastExecutionDate": null,
    "maxAmount": {
      "value": 89000.00,
      "currency": "CZK"
    }
  },
},
"debtorAccount": {
  "identification": {
    "iban": "CZ7508000000002108589434"
  },
  "currency": "CZK"
},
"creditorAccount": {
  "identification": {
    "iban": "CZ6330300000000000000123"
  },
  "currency": "CZK"
},
},

```

```

    "remittanceInformation": {
      "unstructured": "/VS/7418529630/SS/1234567890"
    },
    "signInfo": {
      "state": "OPEN",
      "signId": "164298331754922"
    },
    "instructionStatus": "ACTC"
  }
}

```

5.15.3 Example of response to wrong call

Response 400 Bad Response Headers

```

HTTP/1.1 400 Bad Request
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4

```

Response 400 Bad Response Body

```

{
  "errors": [
    {
      "error": "AC02"
    },
    {
      "error": "DT01",
      "scope": "requestedExecutionDate"
    },
    {
      "error": "AM03",
      "parameters": {
        "CURRENCY": "JPY",
        "CURRENCY_ALLOWED": "CZK"
      }
    }
  ]
}

```



```
    "scope": "debtorAccount.currency"
  }
]
}
```

5.16 Example of request and response for get detail about standing order (AISP and also PISP)

5.16.1 Example of request detail of standing order

Request headers:

```
GET https://api.bank.cz/v1/my/standingorders/033225570000078020091
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:39:01 GMT
User-Involved: true
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456
```

5.16.2 Example of correct response

Response 200 OK Headers:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 200 OK Body

```
{
  "standingOrderIdentification": {
    "instructionIdentification": "SomeID-1285673519924",
    "transactionIdentification": "033225570000078020091"
  },
  "paymentTypeInformation": {
```

```

    "serviceLevel": {
      "code": "DMCT"
    },
    "amount": {
      "instructedAmount": {
        "value": 1500.00,
        "currency": "CZK"
      }
    },
    "requestedExecutionDate": "2018-09-08",
    "standingOrder": {
      "alias": "Payments for lunch - kindergarten Michal",
      "execution": {
        "interval": "MONTHLY",
        "intervalDue": "25",
        "mode": "MAX_AMOUNT_EXCEEDED",
        "modeDue": "DUE_OR_BEFORE_DAY_OF_MONTH"
      },
      "exceptions": {
        "stoppages": ["JULY", "AUGUST"],
        "breaks": [
          {
            "validFromDate": "2018-11-01",
            "validToDate": "2019-01-30"
          },
          {
            "validFromDate": "2019-05-01",
            "validToDate": "2019-06-15"
          }
        ]
      }
    },
    "validity": {

```

```

        "lastExecutionDate": null,
        "maxAmount": {
            "value": 89000.00,
            "currency": "CZK"
        }
    },
    "debtorAccount": {
        "identification": {
            "iban": "CZ7508000000002108589434"
        },
        "currency": "CZK"
    },
    "creditorAccount": {
        "identification": {
            "iban": "CZ6330300000000000000123"
        },
        "currency": "CZK"
    },
    "remittanceInformation": {
        "unstructured": "/VS/7418529630/SS/1234567890"
    },
    "signInfo": {
        "state": "OPEN",
        "signId": "164298331754922"
    },
    "instructionStatus": "ACTC"
}

```

5.16.3 Example of response to wrong call

Response 404 Not Found Headers

```
HTTP/1.1 404 Not Found
```

```
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 404 Not Found Body

```
{
  "errors": [
    {
      "error": "ID_NOT_FOUND",
      "scope": "transactionIdentification"
    }
  ]
}
```

5.17 Example of request and response for get status of standing order

5.17.1 Example of request

Request headers:

```
GET
https://api.bank.cz/v1/my/standingorders/033225570000078020091/status
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:40:01 GMT
User-Involved: false
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456
```

5.17.2 Example of correct response

Response 200 OK Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response **200 OK** Body:

```
{
  "instructionStatus": "ACSP",
  "realizedPayments": [
    {
      "transactionIdentification": "someId56668746523322",
      "date": "2017-01-31T00:00:00.000+01"
    },
    {
      "transactionIdentification": " otherId198568745569",
      "date": "2017-01-31T00:00:00.000+01"
    },
  ]
}
```

5.17.3 Example of response to wrong call

Response **404 Not Found** Headers

```
HTTP/1.1 404 Not Found
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response **404 Not Found** Body

```
{
  "errors": [
    {
      "error": "TRANSACTION_MISSING"
    }
  ]
}
```

5.18 Example of request and response to delete unauthorised and authorised standing order

5.18.1 Example of request

Request headers:

```
DELETE https://api.bank.cz/v1/my/standingorders/033225570000078020091
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:41:01 GMT
User-Involved: true
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456
```

5.18.2 Example of correct response

Response **204 No Content** Headers - in case of unauthorised standing order

```
HTTP/1.1 204 No Content
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response **200 OK** Headers - in case of authorised standing order

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response **200 OK** Body - in case of authorised standing order

```
{
  "standingOrderIdentification": {
    "instructionIdentification": "SomeID-1285673519924",
    "endToEndIdentification": "abcd557000000102",
    "transactionIdentification": "048885570000001020045"
  },
  "signInfo": {
```

```
    "state": "OPEN",
    "signId": "164298331754922"
  }
}
```

5.18.3 Example of response to wrong call

Response 404 Not Found Headers

```
HTTP/1.1 404 Not Found
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 404 Not Found Body

```
{
  "errors": [
    {
      "error": "TRANSACTION_MISSING"
    }
  ]
}
```

5.19 Example of request and response to change existing standing order

5.19.1 Example of request standing order change (Domestic payment)

Request headers:

```
PUT https://api.bank.cz/v1/my/standingorders/033225570000078020091
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:41:01 GMT
User-Involved: true
API-key: 00000000-1212-0f0f-a0a0-123456789abc
```

TPP-Name: Google China

Authorization: Bearer AbCdEf123456

Request Body:

```
{
  "standingOrderIdentification": {
    "instructionIdentification": "SomeID-1285673519924"
  },
  "amount": {
    "instructedAmount": {
      "value": 1650.00,
      "currency": "CZK"
    }
  },
  "requestedExecutionDate": "2018-09-08",
  "standingOrder": {
    "alias": "New payments for lunch - kindergarten Michal",
    "execution": {
      "interval": "MONTHLY",
      "intervalDue": "25",
      "mode": "MAX_AMOUNT_EXCEEDED",
      "modeDue": "DUE_OR_BEFORE_DAY_OF_MONTH"
    },
    "exceptions": {
      "stoppages": ["JULY", "AUGUST"]
    },
    "validity": {
      "lastExecutionDate": null,
      "maxAmount": {
        "value": 45000.00,
        "currency": "CZK"
      }
    }
  }
}
```



```

    }
  },
  "debtorAccount": {
    "identification": {
      "iban": "CZ7508000000002108589434"
    },
    "currency": "CZK"
  },
  "creditorAccount": {
    "identification": {
      "iban": "CZ6330300000000000000123"
    },
    "currency": "CZK"
  },
  "remittanceInformation": {
    "unstructured": "/VS/7418529630/SS/1234567890"
  }
}

```

5.19.2 Example of correct response

Response **200 OK** Headers:

```

HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4

```

Response **200 OK** Body

```

{
  "standingOrderIdentification": {
    "instructionIdentification": "SomeID-1285673519924",
    "transactionIdentification": "033225570000078020091"
  },

```

```
"paymentTypeInformation": {
  "serviceLevel": {
    "code": "DMCT"
  },
  "amount": {
    "instructedAmount": {
      "value": 1650.00,
      "currency": "CZK"
    }
  },
  "requestedExecutionDate": "2018-09-08",
"standingOrder": {
  "alias": "New payments for lunch - kindergarten Michal",
  "execution": {
    "interval": "MONTHLY",
    "intervalDue": "25",
    "mode": "MAX_AMOUNT_EXCEEDED",
    "modeDue": "DUE_OR_BEFORE_DAY_OF_MONTH"
  },
  "exceptions": {
    "stoppages": ["JULY", "AUGUST"]
  },
  "validity": {
    "lastExecutionDate": null,
    "maxAmount": {
      "value": 45000.00,
      "currency": "CZK"
    }
  }
},
"debtorAccount": {
  "identification": {
```

```
        "iban": "CZ7508000000002108589434"
      },
      "currency": "CZK"
    },
    "creditorAccount": {
      "identification": {
        "iban": "CZ633030000000000000123"
      },
      "currency": "CZK"
    },
    "remittanceInformation": {
      "unstructured": "/VS/7418529630/SS/1234567890"
    },
    "signInfo": {
      "state": "OPEN",
      "signId": "fc65339fd01994ac2a657fa97cac37d4"
    },
    "instructionStatus": "ACTC"
  }
}
```

5.19.3 Example of response to wrong call

5.19.3.1 Error 400

Response 400 **Bad Response Headers**

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 400 **Bad Response Body**

```
{
  "errors": [
    {
```

```

    "error": "AC02"
  },
  {
    "error": "DT01",
    "scope": "requestedExecutionDate"
  },
  {
    "error": "AM03",
    "parameters": {
      "CURRENCY": "JPY",
      "CURRENCY_ALLOWED": "CZK"
    },
    "scope": "debtorAccount.currency"
  }
]
}

```

5.19.3.2 Error 404

Response **404 Not Found** Headers

```

HTTP/1.1 404 Not Found
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4

```

Response **404 Not Found** Body

```

{
  "errors": [
    {
      "error": "TRANSACTION_MISSING"
    }
  ]
}

```

5.20 Example of request and response Step I (get authorization) of standing order

5.20.1 Example of request

Request headers:

```
GET
https://api.bank.cz/v1/my/standingorders/033225570000078020091/sign/
fc65339fd01994a

Content-Type: application/json

X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4

Date: Wed, 6 Jan 2019 07:42:01 GMT

User-Involved: true

API-key: 00000000-1212-0f0f-a0a0-123456789abc

TPP-Name: Google China

Authorization: Bearer AbCdEf123456
```

5.20.2 Example of correct response

Response 200 OK Headers

```
HTTP/1.1 200 OK

Content-Type: application/json

X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 200 OK Body:

```
{
  "scenarios": [
    "CODE of The One of the methods ..."
    "CODE of The second method ..."
  ],
  "signInfo": {
    "state": "OPEN",
    "signId": "fc65339fd01994a"
  }
}
```

```
}
```

5.20.3 Example of response to wrong call

5.20.3.1 Error 404

Response 404 Not Found Headers

```
HTTP/1.1 404 Not Found
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 404 Not Found Body

```
{
  "errors": [
    {
      "error": "ID_NOT_FOUND",
      "scope": "signId"
    },
    {
      "error": "ID_NOT_FOUND",
      "scope": "paymentId"
    }
  ]
}
```

5.21 Example of request and response Step II (triggering authorization) of standing order

5.21.1 Example of requests

Request headers:

```
POST
https://api.bank.cz/v1/my/standingorders/033225570000078020091/sign/
fc65339fd01994a
Content-Type: application/json
```

```
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:42:01 GMT
User-Involved: true
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456
```

Request Body only with the selected method:

```
{
  "authorizationType": "SMS"
}
```

Request Body only with the selected method:

```
{
  "authorizationType": "SMS",
  "redirectUrl": "http://my.sometpp.com/..."
}
```

5.21.2 Example of correct response

Response **200 OK** Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response **200 OK** Body:

```
{
  "authorizationType": "USERAGENT_REDIRECT",
  "href": {
    "url": "http://www.bank.cz/authorization/164298331754922",
    "id": "164298331754922"
  },
  "method": "GET",
}
```

```
"formData": {
  "SAMLRequest": "45sa4d4e4e4fds5f6s4df4sd6f",
  "relayState": "token"
},
"signInfo": {
  "state": "OPEN",
  "signId": "fc65339fd01994a"
}
}
```

5.21.3 Example of response to wrong call

5.21.3.1 Error 404

Response **404 Not Found** Headers

```
HTTP/1.1 404 Not Found
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response **404 Not Found** Body

```
{
  "errors": [
    {
      "error": "ID_NOT_FOUND"
    }
  ]
}
```

5.21.3.2 Error 400

Response **400 Bad Request** Headers

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```


Response 400 Bad Request Body

```
{
  "errors": [
    {
      "error": "AUTH_LIMIT_EXCEEDED"
    }
  ]
}
```

5.22 Example of request and response Step III (validate/end authorization) of standing order

5.22.1 Example of request

Request headers:

```
PUT
https://api.bank.cz/v1/my/standingorders/033225570000078020091/sign/
fc65339fd01994a
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:43:01 GMT
User-Involved: true
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456
```

Request Body:

```
{
  "authorizationType": "SMS",
  "oneTimePassword": "12345"
}
```

5.22.2 Example of correct response

Response **200 OK** Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response **200 OK** Body:

```
{
  "state": "DONE",
  "instructionStatus": "ACTC",
  "pollInterval": 5000
}
```

5.22.3 Example of response to wrong call

Response **404 Not Found** Headers

```
HTTP/1.1 404 Not Found
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response **404 Not Found** Body

```
{
  "errors": [
    {
      "error": "ID_NOT_FOUND"
    }
  ]
}
```

5.23 Example of request and response after entering a batch payment

5.23.1 Example of request batch payment initiation

Request headers:

```
POST https://api.bank.cz/v1/my/batchpayments
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:44:01 GMT
User-Involved: true
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456
```

Request Body:

```
{
  "exchangeIdentification": "658576010faf0a23dc",
  "instructionName": "My new batch payment",
  "payments": [
    {
      "signId": "SomeID-1285673519924"
    },
    {
      "signId": "SomeID-1285673519925"
    },
    {
      "signId": "SomeID-1285673519929"
    }
  ]
}
```

5.23.2 Example of correct response

Response 200 OK Headers:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response **200 OK** Body

```
{
  "transactionIdentification": "033225570000078020091"
  "signInfo": {
    "state": "OPEN",
    "signId": "164298331754922"
  }
  "instructionStatus": "ACTC"
}
```

5.23.3 Example of response to wrong call

Response **400 Bad Response** Headers

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response **400 Bad Response** Body

```
{
  "errors": [
    {
      "error": "ACC_DIFFERENT"
    }
    {
      "error": "INCORRECT_SIGNID",
      "parameters": {
        "signID": "SomeID-1285673519924",
        "error": "State CLOSED"
      }
    }
  ]
}
```

5.24 Example of request and response for batch payment delete

5.24.1 Example of request

Request headers:

```
DELETE
https://api.bank.cz/v1/my/batchpayments/033225570000078020091?allInstr
uctions=true
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:45:01 GMT
User-Involved: true
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456
```

5.24.2 Example of correct response

Response 204 No Content Headers

```
HTTP/1.1 204 No Content
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

5.24.3 Example of response to wrong call

Response 404 Not Found Headers

```
HTTP/1.1 404 Not Found
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 404 Not Found Body

```
{
  "errors": [
    {
```

```
    "error": "TRANSACTION_MISSING"
  }
]
}
```

5.25 Example of request and response for batch payment status

5.25.1 Example of request

Request headers:

```
GET https://api.bank.cz/v1/my/batchpayments/eb39d1e4-b718-4011-9fe2-35f025cc0335/status
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:46:01 GMT
User-Involved: false
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456
```

5.25.2 Example of correct response

Response 200 OK Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 200 OK Body:

```
{
  "instructionStatus": "ACSP",
  "payments": [
    {
      "transactionIdentification": "someId56668746523322",
      "signInfo": {
```

```

        "state": "OPEN",
        "signID": "SomeID-1285673519924"
    },
    {
        "transactionIdentification": "someId56668746523322",
        "signInfo": {
            "state": "OPEN",
            "signID": "SomeID-1285673519925"
        }
    },
    {
        "transactionIdentification": "someId56668746523322",
        "signInfo": {
            "state": "OPEN",
            "signID": "SomeID-1285673519929"
        }
    }
]
}

```

5.25.3 Example of response to wrong call

Response **404 Not Found** Headers

```

HTTP/1.1 404 Not Found
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4

```

Response **404 Not Found** Body

```

{
  "errors": [
    {
      "error": "TRANSACTION_MISSING"
    }
  ]
}

```

```
}  
]  
}
```

5.26 Example of request and response to Batch payment authorization Step I

5.26.1 Example of request

Request headers:

```
GET https://api.bank.cz/v1/my/batchpayments/eb39d1e4-b718-4011-  
9fe235f025cc0335/sign/164298331754922  
Content-Type: application/json  
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4  
Date: Wed, 6 Jan 2019 07:46:01 GMT  
User-Involved: true  
API-key: 00000000-1212-0f0f-a0a0-123456789abc  
TPP-Name: Google China  
Authorization: Bearer AbCdEf123456
```

5.26.2 Example of correct response

Response 200 OK Headers

```
HTTP/1.1 200 OK  
Content-Type: application/json  
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 200 OK Body:

```
{  
  "scenarios": [  
    "CODE of The One of the methods ..."  
    "CODE of The second method ..."  
  ],  
  "signInfo": {
```



```
    "state": "OPEN",
    "signId": "164298331754922"
  }
}
```

5.26.3 Example of response to wrong call

Response 404 Not Found Headers

```
HTTP/1.1 404 Not Found
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 404 Not Found Body

```
{
  "errors": [
    {
      "error": "ID_NOT_FOUND",
      "scope": "signId"
    },
    {
      "error": "ID_NOT_FOUND",
      "scope": "paymentId"
    }
  ]
}
```

5.27 Example of request and response to Batch payment authorization Step II

5.27.1 Example of requests

Request headers:

```
POST https://api.bank.cz/v1/my/batchpayments/eb39d1e4-b718-4011-9fe235f025cc0335/sign/164298331754922
```

```
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:47:01 GMT
User-Involved: true
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456
```

Request Body only with the selected method:

```
{
  "authorizationType": "SMS"
}
```

Request Body only with the selected method:

```
{
  "authorizationType": "SMS",
  "redirectUrl": "http://my.sometpp.com/..."
}
```

5.27.2 Example of correct response

Response **200 OK** Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response **200 OK** Body:

```
{
  "authorizationType": "USERAGENT_REDIRECT",
  "href": {
    "url": "http://www.bank.cz/authorization/164298331754922",
    "id": "164298331754922"
  },
}
```

```
"method": "GET",
"formData": {
  "SAMLRequest": "45sa4d4e4e4fds5f6s4df4sd6f",
  "relayState": "token"
},
"signInfo": {
  "state": "OPEN",
  "signId": "164298331754922"
}
}
```

5.27.3 Example of response to wrong call

5.27.3.1 Error 404

Response **404 Not Found** Headers

```
HTTP/1.1 404 Not Found
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response **404 Not Found** Body

```
{
  "errors": [
    {
      "error": "ID_NOT_FOUND"
    }
  ]
}
```

5.27.3.2 Error 400

Response **400 Bad Request** Headers

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 400 Bad Request Body

```
{
  "errors": [
    {
      "error": "AUTH_LIMIT_EXCEEDED"
    }
  ]
}
```

5.28 Example of request and response for batch payment authorization Step III

5.28.1 Example of request

Request headers:

```
PUT https://api.bank.cz/v1/my/batchpayments/eb39d1e4-b718-4011-9fe235f025cc0335/sign/164298331754922
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:47:01 GMT
User-Involved: true
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456
```

Request Body:

```
{
  "authorizationType": "SMS",
  "oneTimePassword": "12345"
}
```

5.28.2 Example of correct response

Response 200 OK Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response **200 OK** Body:

```
{
  "state": "DONE",
  "instructionStatus": "ACTC",
  "pollInterval": 5000
}
```

5.28.3 Example of response to wrong call

Response **404 Not Found** Headers

```
HTTP/1.1 404 Not Found
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response **404 Not Found** Body

```
{
  "errors": [
    {
      "error": "ID_NOT_FOUND"
    }
  ]
}
```

5.29 Example of request and response for list of consents

5.29.1 Example of request for list of consents

Request headers:

```
GET https://api.bank.cz/v1/my/consents
```

```
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:48:01 GMT
User-Involved: false
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456
```

5.29.2 Example of correct response

Response **200 OK** Headers:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response **200 OK** Body

```
{
  "consentId": "b261424c-882a-4ce4-a78a-0f4d664ef5cc",
  "consents": [
    {
      "identification": {
        "iban": "CZ0708000000001019382023",
        "other": "1019382023"
      },
      "accesses": ["AIS.balances", "AISP.transactions", "PIS"],
      "validUntil": "2018-12-24T18:00:00+01:00"
    },
    {
      "identification": {
        "iban": "CZ7408000000001135924571",
        "other": "1135924571"
      },
      "accesses": ["PIS"],
    }
  ]
}
```

```
        "validUntil": "2018-12-24T18:00:00+01:00"
      }
    ]
  }
```

5.29.3 Example of response to wrong call

Response 400 **Bad Response Headers**

```
HTTP/1.1 401 Unauthorised
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 400 **Bad Response Body**

```
{
  "errors": [
    {
      "error": "UNAUTHORISED"
    }
  ]
}
```

5.30 Example of request and response for delete an consent

5.30.1 Example of request for deleting of consents

Request headers:

```
DELETE https://api.bank.cz/v1/my/consents/b261424c-882a-4ce4-a78a-0f4d664ef5cc
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:49:01 GMT
User-Involved: false
API-key: 00000000-1212-0f0f-a0a0-123456789abc
```

```
TPP-Name: Google China
```

5.30.2 Example of correct response

HTTP response example:

```
HTTP/1.1 204 No Content
Date: Sun, 11 Aug 2018 20:03:00 GMT
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

5.30.3 Example of response to wrong call

Response 404 Bad Response Headers

```
HTTP/1.1 404 Not Found
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 404 Bad Response Body

```
{
  "errors": [
    {
      "error": "NOT_FOUND"
    }
  ]
}
```

5.31 Example of request and response for list of authorizations

Query URI used: GET /my/authorizations

5.31.1 Example of request for list of authorizations

Request headers:

```
GET https://api.bank.cz/v1/my/authorizations
Authorization: Bearer AbCdEf123456
```



```
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
Date: Wed, 6 Jan 2019 07:49:01 GMT
User-Involved: false
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
```

5.31.2 Example of correct response

Response **200 OK** Headers:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

Response **200 OK** Body

```
{
  "authorizations": [
    {
      "type": "payment",
      "transactionIdentification": "048885570000001020045",
      "validUntil": "2018-12-24T18:00:00+01:00",
      "scenarios": [
        "CODE of The One of the methods ..."
        "CODE of The second method ..."
      ],
      "signInfo": {
        "state": "OPEN",
        "signId": "164298331754922"
      }
    }
  ]
}
```

```

    }
  },
  {
    "type": "standingorder",
    "transactionIdentification": "71325680000009512001",
    "validUntil": "2018-11-29T10:05:00+01:00",
    "scenarios": [
      "CODE of The One of the methods ..."
      "CODE of The second method ..."
    ],
    "signInfo": {
      "state": "OPEN",
      "signId": "164298331548128"
    }
  }
]
}

```

5.31.3 Example of response to wrong call

Response 400 Bad Response Headers

```

HTTP/1.1 401 Unauthorised
Content-Type: application/json

```

Response 400 Bad Response Body

```

{

```

```
"errors": [  
  {  
    "error": "UNAUTHORISED"  
  }  
]  
}
```

5.32 Example of request and response for Query for List of client's awaiting payments

5.32.1 Example of request

Request headers:

```
GET https://api.bank.cz/v1/my/payments/awaiting  
Content-Type: application/json  
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa9874d5ac2  
Date: Wed, 1 Jan 2020 08:13:01 GMT  
User-Involved: false  
API-key: 00000000-1212-0f0f-a0a0-123456789abc  
TPP-Name: Google China  
Authorization: Bearer AbCdEf123456
```

5.32.2 Example of correct response

Response 200 OK Headers

```
HTTP/1.1 200 OK  
Content-Type: application/json  
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa9874d5ac2
```

Response 200 OK Body

```
{  
  "pageNumber": 0,  
}
```

```

"pageCount": 1,
"nextPage": 0,
"pageSize": 10,
"totalCount": 1,
"payments": [
  {
    "paymentIdentification": {
      "instructionIdentification": "None",
      "transactionIdentification": "CP15125555"
    },
    "paymentTypeInformation": {
      "serviceLevel": {
        "code": "DMCT"
      }
    },
    "amount": {
      "instructedAmount": {
        "value": 3000,
        "currency": "CZK"
      }
    },
    "requestedExecutionDate": "2019-12-22",
    "debtorAccount": {
      "id": "CUA1234567899",
      "identification": {
        "iban": "CZ8858000000001234567899",
        "other": {
          "identification": "1234567899"
        }
      }
    },
    "creditorAgent": {

```

```

    "financialInstitutionIdentification": {
      "clearingSystemMemberIdentification": {
        "clearingSystemIdentification": {
          "code": "0100"
        }
      }
    },
    "creditorAccount": {
      "identification": {
        "other": {
          "identification": "000019-2784000277"
        }
      }
    },
    "remittanceInformation": {
      "structured": {
        "creditorReferenceInformation": {
          "reference": [
            "VS:20200312"
          ]
        }
      }
    },
    "awaitingStatus": "FUTURE"
  }
]
}

```

5.32.3 Example of response to wrong call

5.32.3.1 Error 404

Response 404 Not Found Headers

```
HTTP/1.1 404 Not Found
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 404 Not Found Body

```
{
  "errors": [
    {
      "error": "ID_NOT_FOUND"
    }
  ]
}
```

5.32.3.2 Error 400

Response 400 Bad Request Headers

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4
```

Response 400 Bad Request Body

```
{
  "errors": [
    {
      "error": "AM03",
      "scope": "currency"
    },
    {
      "error": "DT01",
      "parameters": {
        "DATE": "DATE_TO_OLD"
      }
    }
  ]
}
```

```

    },
    "scope": "fromDate"
  },
  {
    "error": "DT01",
    "parameters": {
      "DATE": "DATE_IN_FUTURE"
    },
    "scope": "toDate"
  }
]
}

```

5.33 Example of request and response ROS registration

5.33.1 Example of payment response with authorization object

Response 200 OK Headers

```

HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa94fbd51a4

```

Response 200 OK Body

```

{
  "paymentIdentification": {
    "instructionIdentification": "NOTPROVIDED",
    "transactionIdentification": "048885570000001020045"
  },
  "paymentTypeInformation": {
    "instructionPriority": "NORM",
    "serviceLevel": {
      "code": "DMCT"
    }
  }
}

```

```

    }
  },
  "amount": {
    "instructedAmount": {
      "value": 10050.15,
      "currency": "CZK"
    }
  },
  "requestedExecutionDate": "2017-02-20",
  "debtorAccount": {
    "identification": {
      "iban": "CZ0708000000001019382023"
    },
    "currency": "CZK"
  },
  "creditorAccount": {
    "identification": {
      "iban": "CZ0708000000001019540081"
    },
    "currency": "CZK"
  },
  "remittanceInformation": {
    "structured": {
      "creditorReferenceInformation": {
        "reference": [
          "VS:501",
          "KS:9",
          "SS:1005"
        ]
      }
    }
  },
},

```



```

    "signInfo": {
      "state": "OPEN",
      "signId": "164298331754922"
    },
    "instructionStatus": "ACWC",
    "statusChangeInfo": "DT01",
    "structured_scope": {
      "authorizationObject":
        "eyJhbGciOiJSUzI1NiIsImtpZCI6ImN5YmRjIn0.ew0KICJpc3MiOiA
        (... abbreviated for brevity ...)
        zCYIb_NMXvtTIVcljpspnTSD7xMbpL-2QgwUsAlMGzw"
    }
  }
}

```

5.33.2 Example of request

Request headers:

```

POST https://api.bank.cz/v1/ros
Content-Type: application/jwt
X-Request-ID: 2258fffc-1587-44d4-9f2b-3aa9874d6658
Date: Wed, 1 Apr 2021 09:15:02 GMT
User-Involved: false
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China

```

Response Body

```

eyJhbGciOiJSUzI1NiIsImtpZCI6ImN5YmRjIn0.ew0KICJpc3MiOiA
(... abbreviated for brevity ...)
zCYIb_NMXvtTIVcljpspnTSD7xMbpL-2QgwUsAlMGzw

```

Example of JWT content (in abbreviated form):

Example of JWT content for the request to /ros endpoint payment registration. The object includes elements of a common / auth call, such as client_id and redirect_uri. But also the payment instruction itself, see the payment element.

According to COBS, the content of the payment element is identical to the payload from the initiation of payment in the 5.0 standard.

```
{
  "iss": "s6BhdRkqt3",
  "aud": "https://api.bank.cz",
  "response_type": "code",
  "client_id": "someClient001",
  "redirect_uri": "https://client.sometpp.cz/cb",
  "state": "af0ifjsldkj",
  "structured_scope": {
    "authorizationObject":
      "eyJhbGciOiJSUzI1NiIsImtpZCI6Im5yYmRjIn0.ew0KICJpc3MiOiA
      (... abbreviated for brevity ...)
      zCYIb_NMXvtTIVcljpspnTSD7xMbpL-2QgwUsAlMGzw"
  }
}
```

5.33.3 Example of correct response

Response 201 OK Headers

```
HTTP/1.1 201 Created
Content-Type: application/json
X-Request-ID: 2258fffc-1587-44d4-9f2b-3aa9874d6658
```

Response 201 OK Body

```
{
```

```

    "iss": "https://app.sometpp.cz/",
    "aud": "s6BhdRkqt3",
    "request_uri": "urn:example:QVh0fccba",
    "exp": 1493738581
  }

```

5.33.4 Example of correct IDToken

```

{
  "iss": "s6BhdRkqt3",
  "aud": "https://api.bank.cz/ros",
  "exp": 1617228000,
  "auth_time": 1617208000,
  "jti": "61095178-06cc-437a-bc63-c6ef9cda571b:PISP",
  "structured_scope": {
    "authorizationObject":
      "eyJhbGciOiJSUzI1NiIsImtpZCI6Im9yYmRjIn0.ew0KICJpc3MiOiA
      (... abbreviated for brevity ...)
      zCYIb_NMXvtTIVcljpspnTSD7xMbpL-2QgwUsAlMGzw"
  }
}

```

5.34 Example of request and response overview of unsecured loans initiated via TPP

5.34.1 Example of request

Request headers:

```

GET https://api.bank.cz/v1/my/loans/bd089c1b-67dd-40fd-a9de-
0c66fb47af8c
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa987785ac2

```

```
Date: Wed, 1 Jan 2021 09:25:01 GMT
User-Involved: false
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456
```

5.34.2 Example of correct response

Response 200 OK Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 2258fffc-1587-44d4-9f2b-3aa9874d7785
```

Response payload

```
{
  "TPPName": "Cool FinTech",
  "loans": [
    {
      "loanApplicationID": "5e65a492-ed25-4b90-b3aa-2f9127765659",
      "amount": {
        "value": 10500,
        "currency": "CZK"
      },
      "from": "2021-06-03",
      "status": "APPROVED"
    },
    {
      "loanApplicationID": "10e012d3-9f96-4ed0-b3c1-7e5930879c55",
      "amount": {
        "value": 10000,
        "currency": "CZK"
      },
    },
  ]
}
```

```
        "from": "2021-04-03",
        "status": "DRAWN"
    }
]
}
```

5.35 Example of request and response loan offers (GET/my/loanoffers{?loanPeriod,amount,currency})

5.35.1 Example of request

Request headers:

```
GET https://api.bank.cz/v1/my/loanoffers?amount=50000
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa987785ac2
Date: Wed, 1 Jan 2021 09:25:01 GMT
User-Involved: false
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456
```

5.35.2 Example of correct response

Response 200 OK Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 2258fffc-1587-44d4-9f2b-3aa9874d8854
```

Response payload

```
{
  "loanExample": [
    {
      "amount": {
        "value": 60000,
        "currency": "CZK"
      }
    }
  ]
}
```

```

    },
    "interestRate": "7.9",
    "apr": "9.1",
    "monthlyPayment": "921",
    "totalPaments": "15",
    "totalFee": "500",
    "totalRepaymentAmount": "70000"
  },
  {
    "amount": {
      "value": 50000,
      "currency": "CZK"
    },
    "interestRate": "6.9",
    "apr": "8.1",
    "monthlyPayment": "750",
    "totalPaments": "15",
    "totalFee": "500",
    "totalRepaymentAmount": "65000"
  }
]
}

```

5.36 Example of request and response loan application (POST /my/loans)

5.36.1 Example of request

Request headers:

```

POST https://api.bank.cz/v1/my/loans
Content-Type: application/json
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa987785ac2

```

```
Date: Wed, 1 Jan 2021 09:25:01 GMT
User-Involved: false
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456
```

Request payload

```
{
  "loanPeriod": "15",
  "amount": {
    "value": "25000",
    "currency": "CZK"
  },
  "account": "CZ00001247888001245",
  "callbackURL": "https://www.sometpp.cz/loancallback"
}
```

5.36.2 Example of callback response (on specified callbackURL)

Response 302 OK Headers

```
HTTP/1.1 302 Found
Location: https://www.sometpp.cz/post_logout/loancallback&state=${state}&
        loanApplicationID=f03c0f7a-ca39-4f9b-8a4e-298dd568983b
X-Request-ID: 2258fffc-1587-44d4-9f2b-3aa9874d8854
```

5.37 Example of request and response loan application status (GET /my/loans/{loanApplicationID}/status)

5.37.1 Example of request

Request headers:

```
GET https://api.bank.cz/v1/my/loans/f03c0f7a-ca39-4f9b-8a4e-
298dd568983b/status
Content-Type: application/json
```

```
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa987799568
Date: Wed, 1 Jan 2021 09:25:01 GMT
User-Involved: false
API-key: 00000000-1212-0f0f-a0a0-123456789abc
TPP-Name: Google China
Authorization: Bearer AbCdEf123456
```

5.37.2 Example of correct response

Response 200 OK Headers

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-ID: 2258fffc-1587-44d4-9f2b-3aa9874d8854
```

Response payload

```
{
  "loanApplicationID": "f03c0f7a-ca39-4f9b-8a4e-298dd568983b",
  "from": "2021-02-25",
  "amount": {
    "value": "25000",
    "currency": "CZK"
  },
  "status": "DRAWN",
  "account": "CZ00001247888001245"
}
```

5.38 Example of request and response initiation of drawing a loan (POST /my/loans/{loanApplicationID}/draw)

5.38.1 Example of request

Request headers:


```
POST https://api.bank.cz/v1/my/loans/f03c0f7a-ca39-4f9b-8a4e-298dd568983b/draw
```

```
Content-Type: application/json
```

```
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa987799568
```

```
Date: Wed, 1 Jan 2021 09:25:01 GMT
```

```
User-Involved: false
```

```
API-key: 00000000-1212-0f0f-a0a0-123456789abc
```

```
TPP-Name: Google China
```

```
Authorization: Bearer AbCdEf123456
```

5.38.2 Example of correct response

Response 200 OK Headers

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
```

```
X-Request-ID: 2258fffc-1587-44d4-9f2b-3aa9874d8854
```

5.39 Example of request and response loan application cancellation (DELETE/my/loans/{loanApplicationID})

5.39.1 Example of request

Request headers:

```
DELETE https://api.bank.cz/v1/my/loans/f03c0f7a-ca39-4f9b-8a4e-298dd568983b
```

```
Content-Type: application/json
```

```
X-Request-ID: 55d4fffc-2634-44d4-9f2b-3aa987799568
```

```
Date: Wed, 1 Jan 2021 09:25:01 GMT
```

```
User-Involved: false
```

```
API-key: 00000000-1212-0f0f-a0a0-123456789abc
```

```
TPP-Name: Google China
```

```
Authorization: Bearer AbCdEf123456
```

5.39.2 Example of correct response

Response 200 OK Headers

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
```

```
X-Request-ID: 2258fffc-1587-44d4-9f2b-3aa9874d8854
```