

Swiss Corporate API

Table of Contents

1. Overview	1
1.1. Version information.....	1
1.2. Contact information.....	1
2. General	2
2.1. Swiss Corporate API.....	2
2.2. Documentation Structure.....	2
2.3. Version Control	3
2.4. Design Principles	3
2.5. Extensible Architecture.....	3
2.6. References	4
2.6.1. Documents	4
2.6.2. Organizations.....	5
3. Use Cases	6
3.1. "AIS": Account Information Service - third party access to FI account data	6
3.1.1. Retrieve list of authorized accounts (/accounts)	6
3.1.2. Retrieve information about a single specific account (/accounts/{accountId})	6
3.1.3. Retrieve account balance information (/accounts/{accountId}/balance)	6
3.1.4. Retrieve transactions of a specific account (/accounts/{accountId}/transactions)	6
3.2. "PIS": Payment Initiation Service - third party payment instruction submission.....	6
3.2.1. Submit payment instructions (/payments)	6
3.2.2. Retrieve a payment submission (/payments/{submissionId})	7
3.2.3. Retrieve payment submission status information (/payments/{submissionId}/status)	8
3.3. "SPSSTMT": Transport XML account statements.....	9
3.3.1. Retrieve a list of resource links to account statements (/iso20022/statements)	9
3.4. "SPSPMTS": Transport XML payment initiations	9
3.4.1. Submit an ISO20022 XML payment instruction PAIN.001 (/iso20022/payments)	9
3.4.2. Retrieve ISO20022 XML status report PAIN.002 of a payment instruction (/iso20022/payments/{submissionId})	9
4. Conventions	10
4.1. Character set	10
4.1.1. Character set for references	10
4.2. Date and Time Formats	10
4.3. Customer references	10
4.4. Payment Types	11
4.5. Data Model payment initiation submission (PIS)	11
4.5.1. Submit payment instruction	11
4.6. Data Model access to account information (AIS)	13
4.6.1. Request list of accounts	13

4.6.2. Request account information.....	15
4.6.3. Request account balance.....	17
4.6.4. Request account transactions	19
4.7. Mapping to Schemes and Standards	21
4.7.1. Swiss Corporate API attributes vs. SPS ISO-20022.....	21
5. Resources	23
5.1. Payments	23
5.1.1. Submit payment instructions.....	23
5.1.2. Retrieve a payment submission	25
5.1.3. Retrieve payment submission status information	27
5.2. Accounts	28
5.2.1. Retrieve list of authorized accounts.....	29
5.2.2. Retrieve information about a single specific account	31
5.2.3. Retrieve account balance information	33
5.2.4. Retrieve transactions of a specific account	36
5.3. Iso20022	45
5.3.1. Submit an ISO20022 XML payment instruction (PAIN.001)	45
5.3.2. Retrieve the submitted ISO20022 XML PAIN.001 message.	47
5.3.3. Retrieve ISO20022 XML status report (PAIN.002) of a payment instruction	49
5.3.4. Retrieve a list of resource links to account statements (CAMT.053)	51
5.3.5. Retrieves the designated ISO20022 XML CAMT document.	53
5.4. Directory	55
5.4.1. Retrieve a list of all registered clients (e.g., third party providers)	55
5.4.2. Retrieve the information for a registered client (e.g. third party provider).....	57
5.4.3. Retrieve a list of all registered providers (e.g., financial institutions).....	59
5.4.4. Retrieve the informatoin for a registered provider (e.g., financial institutions)	60
5.5. Consents	62
5.5.1. Retrieve a list of consents	62
5.6. Oauth	65
5.6.1. Get the OAuth access and refresh token	66
5.7. Diagnostics	68
5.7.1. Returns all specified request headers and additional diagnostic information	68
5.7.2. Returns all specified request headers, the request body and additional diagnostic information	69
6. Definitions	71
6.1. paymentSubmissionRequest	71
6.2. paymentInstructionItem.....	71
6.3. paymentIBANDetail	72
6.4. paymentISRDetail	72
6.5. paymentOtherDetail	72
6.6. paymentCreditor.....	73

6.7. paymentChargeBearerMethod	73
6.8. paymentBookingInstruction	73
6.9. paymentSubmissionStatus	73
6.10. paymentInstructionItemStatus	73
6.11. paymentCreditorAgent	74
6.12. paymentIbanAccount	74
6.13. paymentOtherAccount	74
6.14. balanceCurrencyAmount	74
6.15. paymentCurrencyAmount	75
6.16. paymentIbanRemittanceReference	75
6.17. paymentIsrRemittanceReference	75
6.18. paymentOtherRemittanceReference	75
6.19. paymentRemittanceInformation	76
6.20. paymentEndToEndId	76
6.21. accountItem	76
6.22. accountBalanceItem	76
6.23. accountTransactionReport	77
6.24. accountTransactionEntry	77
6.25. accountTransactionItem	78
6.26. accountTransactionItemCharges	79
6.27. accountTransactionItemChargesRecord	79
6.28. accountTransactionCounterparty	80
6.29. accountTransactionCounterpartyAgent	80
6.30. accountTransactionInstructedAmount	80
6.31. accountTransactionBankTransactionCode	81
6.32. accountTransactionTransactionType	81
6.33. accountTransactionCounterpartyAccount	81
6.34. accountTransactionRemittanceReference	81
6.35. consentItem	81
6.36. consentAccountItem	82
6.37. directoryClientItem	82
6.38. directoryProviderItem	83
6.39. directoryParticipantItem	84
6.40. directoryContact	85
6.41. directorySoftwareDescription	85
6.42. directoryMarketingDescription	86
6.43. directoryShortDescription	86
6.44. directoryUseCaseItem	86
6.45. directoryProductName	87
6.46. directoryMultisizeImageReference	87
6.47. directoryParticipantStatus	87

6.48. directoryImageReference	87
6.49. iso20022ReportReference	87
6.50. oauthTokenResponse	88
6.51. commonClearingSystemMemberIdentification	88
6.52. commonStructuredOrUnstructuredAddress	88
6.53. commonStructuredAddress	88
6.54. commonUnstructuredAddress	89
6.55. commonErrorResponse	89
6.56. commonErrorType	89
6.57. healthCheckResponse	90
7. Security and Access Control	91
7.1. Transport Layer Security	91
7.2. Authentication	91
7.3. High Level Authorization	91
7.4. Consent Management	91
7.5. Request validation	96
7.6. Token refresh	96
8. Participants Directory	101
8.1. Access	101
8.2. Management	101
8.3. Content	101
8.3.1. Participants	101
8.3.2. Software	102
8.3.3. Use cases	102
8.4. Deliverables by participants	102
8.4.1. Additional deliverables for TPPs	103
8.4.2. Additional deliverables for FIs	103
9. Glossary	105
10. Appendix	106
10.1. Reason Codes	106
10.2. Allowed Certificates for platform access	108
10.3. Changelog	108
10.4. Problem Definitions	109
10.4.1. Bad Request (400)	109
10.4.2. Unauthorized (401)	110
10.4.3. Forbidden (403)	110
10.4.4. Not Found (404)	110
10.4.5. Not Allowed (405)	110
10.4.6. Technical Errors (500)	111
10.5. Open Points	111

Chapter 1. Overview

This is the **release candidate** version of the "Swiss Corporate API" specification. Since this is an **release candidate** version, there are still some open points and ideas which are up for discussion.

Some points are specially marked as **TBD**: for "to be discussed/defined".

Do you have any feedback for us? We'd love to hear about it.

1.1. Version information

Version : 1.0.0.2-SNAPSHOT

1.2. Contact information

Contact : SIX Group

Contact Email : swisscorpapi@six-group.com

Chapter 2. General

2.1. Swiss Corporate API

The goal of the Swiss Corporate API platform is to provide an unified Swiss infrastructure by SIX for any interactions between third party providers (TPPs) and financial institutions.

The first version of the Swiss Corporate API platform focuses on an optimized solution for cloud-based business software (e.g.accounting) providers who want to seamlessly integrate their customers' bank accounts through their software solutions.

Currently, every TPP willing to integrate with a financial institution has to cope with the different provided APIs and protocols provided by each financial institution (FI), facing a range of challenges.

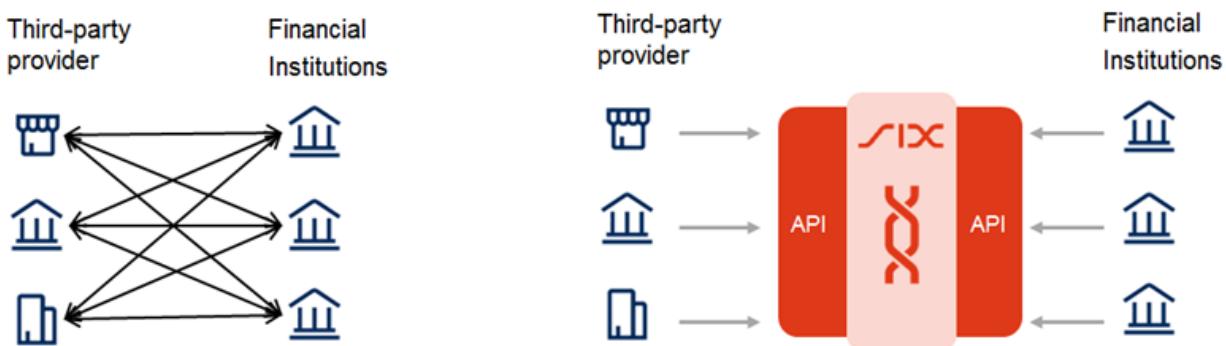


Figure 1. *n:n* connections without a central service vs. reduced number of connections needed with a central service

2.2. Documentation Structure

This documentation consists of the following sections:

[\[Introduction\]](#)

Provides an overview of the scope of the API and the key decisions and principles that contributed to the specification.

[Use Cases](#)

Contains a high level description of the available use cases.

[Conventions](#)

Defines the character set and provides information about the references used within the messages.

Additionally, it contains the data models for the API operations.

[Resources](#)

Describes the endpoints of the API generated from the Swagger definition and enriched with examples for JSON requests and responses. The examples are also provided in the ZIP-File of the documentation.

Security and Access Control

Specifies the means for a TPP to authenticate itself and for a user to authorize access.

Participants Directory

Describes the details to the participant directory of the Swiss Corporate API, containing financial institutions and TPPs.

Glossary

Defines and explains relevant abbreviations.

Appendix

Additional information (i.e. changelog, list of ISO reason codes).

2.3. Version Control

SIX reserves the right to amend this documentation as required within the scope of the applicable contractual conditions.

All rights are reserved with respect to this documentation, including with regard to photocopying and storage on electronic media as well as translation into other languages.

This documentation has been compiled with the greatest care, but may nevertheless contain errors or inaccuracies. SIX cannot assume any legal responsibility or any liability for erroneous information or its consequences.

If you notice any errors in this documentation or have any suggestions for improvements, we would be grateful to receive your feedback by email to swisscorpapi@six-group.com.

The changes made between one published version and another, are tracked in the appendix.

2.4. Design Principles

The API sticks to RESTful API concepts as closely as possible and as long as it is reasonable. However, the main focus lies on having an API that stands out by its simplicity and ease of use. Wherever the RESTful principles would be convoluted and complex, the principles have not been followed.

The Interface Description Language used is the Swagger Specification version 2.0, also known as Open API : <http://swagger.io/>

2.5. Extensible Architecture

The first version of the Swiss Corporate API caters for the use cases described below, but the architecture can easily be extended by additional use cases in the future.

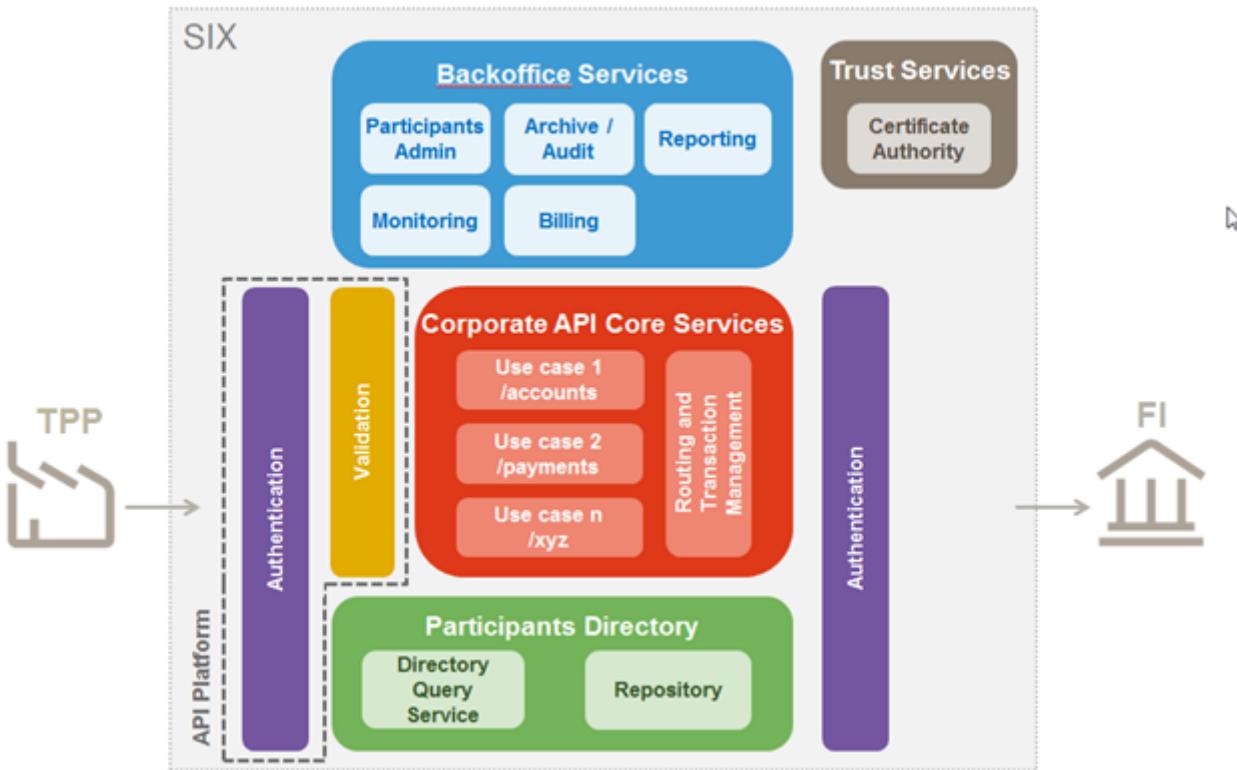


Figure 2. Visualization of extensible Swiss Corporate API architecture

2.6. References

2.6.1. Documents

Reference	Document	Title	Source
[1]	pain.001.001.03	XML Schema Customer Credit Transfer Initiation V03	ISO
[2]	pain.002.001.03	XML Schema Customer Payment Status Report V03	ISO
[3]	camt.053.001.04	XML Schema Bank To Customer Statement V04	ISO
[5]	Swiss Payment Standards - Standardization	Schemes and Examples	SIX Interbank Clearing
[6]	Swiss Implementation Guidelines for Credit Transfers	Swiss Implementation Guidelines for Customer-Bank Messages Credit Transfer (Payment Transactions), Version 1.8	SIX Interbank Clearing

[7]	Swiss Implementation Guidelines for Status Report	Swiss Implementation Guidelines for Customer-Bank Messages for Status Report	SIX Interbank Clearing
[8]	Swiss Implementation Guidelines for Cash Management	Swiss Implementation Guidelines for Customer-Bank Messages (Reports), Version 1.6	SIX Interbank Clearing
[9]	Swiss Business Rules	ISO 20022 Payments - Swiss Business Rules for Payments and Cash Management for Customer-Bank Messages	SIX Interbank Clearing
[10]	RFC 6749	The OAuth 2.0 Authorization Framework	Internet Engineering Task Force (IETF)
[11]	RFC 5785	Defining Well-Known Uniform Resource Identifiers (URIs)	Internet Engineering Task Force (IETF)
[12]	IETF Draft	OAuth 2.0 Authorization Server Metadata	Internet Engineering Task Force (IETF)

2.6.2. Organizations

Organization	Link
SIX	www.six-group.com
ISO	www.iso20022.org
SIX Interbank Clearing	www.iso-payments.ch
Swiss Payment Standards	www.paymentstandards.ch
Internet Engineering Task Force (IETF)	www.ietf.org

Chapter 3. Use Cases

3.1. "AIS": Account Information Service - third party access to FI account data

3.1.1. Retrieve list of authorized accounts (`/accounts`)

This request can be used by a user to retrieve a list of authorized accounts for the specified TPP by the FI. Authorized accounts are defined as those accounts of a user for which a consent has been granted to the TPP to access these accounts for balances or transactions.

3.1.2. Retrieve information about a single specific account (`/accounts/{accountId}`)

This request can be used to retrieve detailed information about one specific authorized account. As a result, the TPP will receive the account identification, IBAN, designation and currency for the specific authorized account.

If the user has granted access (to the TPP) to multiple accounts, then a corresponding transaction has to be submitted for each account separately.

3.1.3. Retrieve account balance information (`/accounts/{accountId}/balance`)

This request can be used to receive the balance (closing booked or interim available) for a specific account. As a result, the TPP will receive the balance (closing booked or interim available) for the account identified in the request.

If the user has granted access (to the TPP) to multiple accounts, then a corresponding transaction has to be submitted for each account separately.

3.1.4. Retrieve transactions of a specific account (`/accounts/{accountId}/transactions`)

The TPP can use this request to receive information about account transactions of a specific account. As a result, the TPP will receive information about all payment transactions booked during the day indicated in the request.

If the user has granted access (to the TPP) to multiple accounts, then a corresponding request has to be submitted for each account separately.

3.2. "PIS": Payment Initiation Service - third party payment instruction submission

3.2.1. Submit payment instructions (`/payments`)

This request can be used to initiate a collective payment in form of a credit transfer from an

authorized account of the payer to an account of the payee. With regard to collective payments, following has to be taken into consideration:

- A collective payment contains transactions for one account, one currency and for one value date.
- Collective payments can contain transactions of different payment types.
- A single payment is a collective payment with only one transaction.

If the message is valid, a resource is created.

The following diagram shows the sequence for the submission of a payment instruction (happy flow):

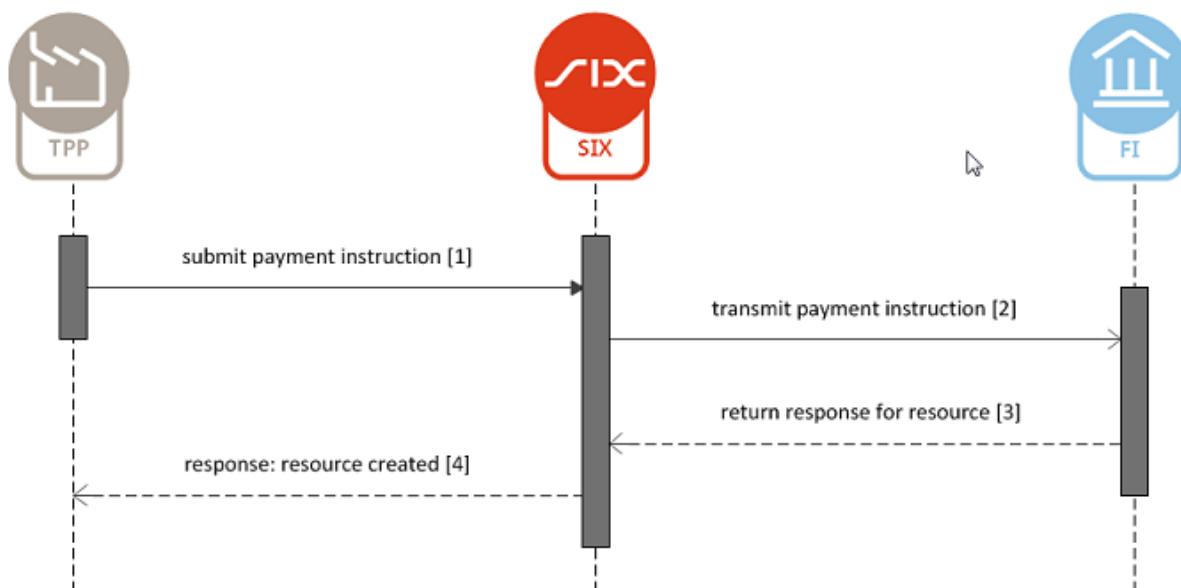


Figure 3. Submission of a payment instruction

1. The TPP submits the payment instruction.
2. The payment instruction is transmitted to the FI of the payer.
3. The FI creates the resource for the payment instruction and returns the response including a unique id (submissionId).
4. The TPP receives the response with the unique id (submissionId) for the payment instruction. The submissionId can be used by the TPP to retrieve the submitted payment instruction and its status.

In case the resource cannot be created (e.g. invalid format of the request) the FI responds with an error code.

3.2.2. Retrieve a payment submission (/payments/{submissionId})

The TPP can retrieve a payment instruction. As a result, the TPP will receive the submitted payment instruction for the resource indicated in the request.

Additionally, a link to the status resource is returned.

3.2.3. Retrieve payment submission status information (/payments/{submissionId}/status)

The TPP can retrieve the status of a submitted payment instruction (only if TPP is initiating party) by using the resource created after submission.

The status of the payment instruction is final. A payment instruction can have following status:

- Accepted (ACCP)
- Partially accepted (PART)
- Rejected (RJCT)

If a payment instruction is partially accepted or rejected, the FI must return the rejected transaction(s) for the payment instruction with additional status information.

The following diagram shows the sequence for the retrieval of the payment instruction status (happy flow):

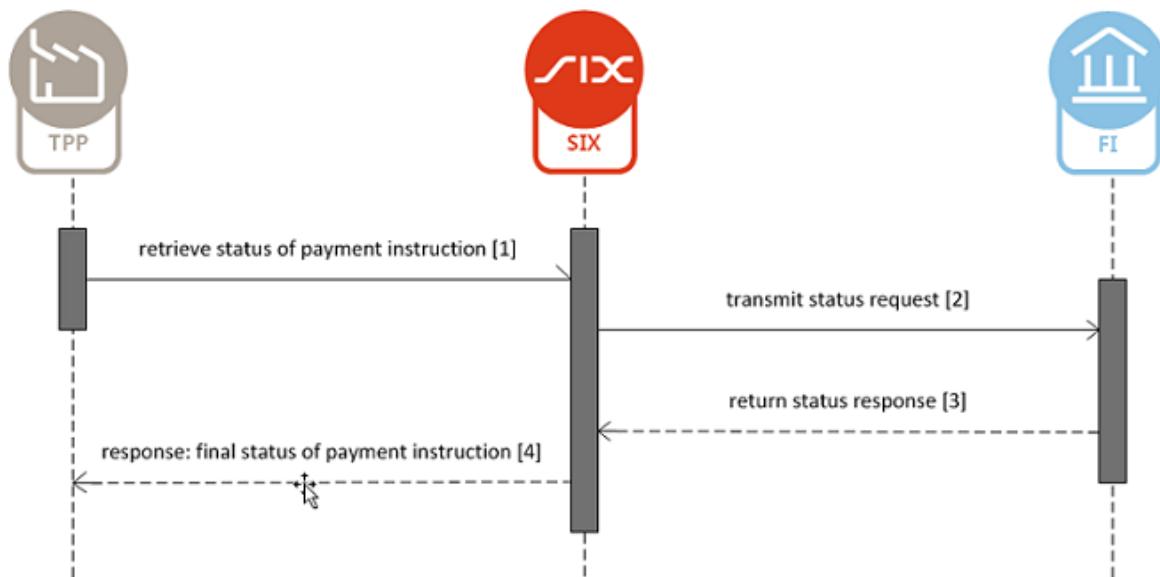


Figure 4. Retrieve payment instruction status information

1. The TPP submits the request to retrieve the status of the payment instruction.
2. The request is transmitted to the FI of the payee.
3. The FI returns the response.
4. The TPP receives the response with the final status of the submitted payment instruction.

In case the status of the payment instruction is not yet available, the FI responds with an error code. The TPP can reuse the resource to poll for the status.

Important: The status message of the corresponding payment instruction must be available for seven calendar days after submission.

3.3. "SPSSTMT": Transport XML account statements

3.3.1. Retrieve a list of resource links to account statements (/iso20022/statements)

To support the Swiss Payment Standard ISO-20022 files implemented by financial institutions, a separate channel will be offered for receiving camt.053 ([3]).

The financial institutions supporting this use case, will offer a location where these files can be downloaded as configured in the clients master data.

The financial institutions providing this location, will only offer end-of-day statements. For intra-day statements, please refer to /accounts/{accountId}/transactions.

3.4. "SPSPMTS": Transport XML payment initiations

3.4.1. Submit an ISO20022 XML payment instruction PAIN.001 (/iso20022/payments)

To support the Swiss Payment Standard ISO-20022 files implemented by the FIs, a separate channel will be offered for submitting pain.001.

3.4.2. Retrieve ISO20022 XML status report PAIN.002 of a payment instruction (/iso20022/payments/{submissionId})

A unique id (submissionId) has to be assigned by the FI for every pain.001 submission. This id can then be used to retrieve the corresponding pain.002 message.

Chapter 4. Conventions

4.1. Character set

In principle, the API supports the UTF-8 character set. The exact list of permitted characters and a conversion table can be found in the Swiss Implementation Guidelines [9].

If characters are sent that are not specified, the message is rejected.

4.1.1. Character set for references

For certain references, only characters from the SWIFT character set are permitted:

- Message Identification
- Payment Information Identification
- End To End Identification

Furthermore, these references must not begin with "/" and must not contain "//".

It is recommended to avoid the use of spaces in the "Message Identification" and "Payment Information Identification" references.

4.2. Date and Time Formats

All dates and timestamps in the JSON payloads are represented in ISO 8601 format matching the following patterns

- Date: yyyy-MM-dd
- Date-time: yyyy-MM-ddTHH:mm:ss+HH:mm
- Date-time (only UTC): yyyy-MM-ddTHH:mm:ssZ

All date-time fields must use local time with offset to UTC. An example is: **2017-04-05T10:43:07+02:00** which indicates a 2 hour offset to UTC.

Zero offset, i.e., UTC, is represented either with a Z or with +00:00, e.g., **2017-04-05T10:43:00+00:00** or **2017-04-05T10:43:00Z**.

Note: At the moment date with time is only used for the healthcheck and directory endpoints. All other definitions use just the date.

4.3. Customer references

In addition to the references mentioned above in the processing chain, a customer reference can also be sent in the Remittance Information, in structured or unstructured form.

Using the Swiss ISR reference

In Switzerland the ISR reference enables the creditor to make automatic comparisons between bills

and the incoming payments.

Using the Swiss QR reference

In Switzerland the QR reference allows possible the automatic comparison between bills and incoming payments for the creditor. The QR reference corresponds to the ISR reference number.

Using the ISO Creditor reference

The ISO Creditor reference (ISO 11649) enables the creditor to make automatic comparisons between bills and the incoming payments.

End to End Identification

End To End Identification is used for the unique identification of a transaction and is assigned by the debtor. The End To End Identification (e.g. the order number) is passed unchanged along the complete processing chain.

4.4. Payment Types

The payment types defined in the Swiss Business Rules [9] are considered for this API. For the messages regarding the payment initiation submission and the request of account transactions, the payment types are categorized as follows:

- **IBAN payments:**

Collection for IBAN payments with following characteristics:

- contains a valid swiss creditor and swiss debtor IBAN (CH or LI), currency CHF or EUR (PT3).
- contains a valid swiss creditor and swiss debtor IBAN (CH or LI), currency is not CHF or EUR (PT4).
- contains a valid SEPA-country creditor and swiss debtor IBAN (CH or LI), currency is EUR (PT5)
- contains a foreign non SEPA-country creditor and swiss debtor IBAN (CH or LI), contains a creditor agent, all currencies allowed (PT6).

- **ISR payments:**

Contains a valid swiss debtor IBAN (CH or LI) an ISR-participant number, creditor agent is empty, ISR-reference is delivered, currency is CHF or EUR.

- **Other payments:**

Collection for all other payments with following characteristics:

- contains a domestic or foreign creditor bank account, swiss debtor IBAN (CH or LI), contains a creditor agent, all currencies allowed.

4.5. Data Model payment initiation submission (PIS)

4.5.1. Submit payment instruction

Corporate API - /payments - request								
Status	Message Item	Technical	IBAN	ISR	Other	Length	Description	Example
M	Message Identification	messageId	X	X	X	35	Format [UUID 32l], formatted as IETF's RFC 4122 v4 lowercase, no hyphen]	eb6305c9117f49deae016487c27b42d
M	Initiating Party Identification	initiatingPartyId	X	X	X	32	Contains an identifier of the TPP who created the message derived from the OAuth 2.0 flow.	TPP01746
M	Requested Execution Date	requestedExecutionDate	X	X	X	10	Contains the requested execution date. In some cases, the financial institution can/will change this date, if the requested date is a bank / post holiday. If no date is chosen, the receiving financial institution will execute on the next possible date.	2018-04-07
M	Debtor Account	debtorAccount						
M	+Type	type		X	X	4	Allowed account identification type for the debtor account. The debtor account must always be an IBAN.	IBAN
M	+Identification	identification		X	X	34	Contains the debtors account (payer) in IBAN format. [A-Z]{2,3}[0-9]{2,2}[a-zA-Z0-9]{1,30}	CH3908704016075473007
M	Booking Instruction	bookingInstruction		X	X	X	21 Permitted codes are: • SINGLEBOOKING_SIA: To be booked as one booking with a single advice • SINGLEBOOKING_NOA: To be booked as one booking with no advice • BATCHBOOKING_SALA_NOA: To be booked as a salary batch with no advice • BATCHBOOKING_SALA_CND: To be booked as salary batch with a collective advice no details • BATCHBOOKING_NOA: To be booked as batch with no advice • BATCHBOOKING_CND: To be booked as batch with a collective advice no details • BATCHBOOKING_CWD: To be booked as batch with a collective advice with details	
M	Transactions	transactions						
M	+Instruction Identification	instructionId	X	X	X	35	Contains an unique tpp-defined ID for the transaction on the list.	tx123456789
M	+End-to-End Identification	endToEndId	X	X	X	35	Contains an unique debtor (payer) reference. The identifier is forwarded to the creditor (payee).	ENDTOENDID-001
M	+Instructed Amount	instructedAmount						
M	+Amount	amount	X	X	X	18 (12.5)	Contains the amount to be paid. Amount must be between 0.01 and 999999999.99.	284.70
M	++Currency	currency		X	X	X	3 payment type 1, 3: only <CHF> or <EUR> permitted payment type 4: all currencies (depending on financial institution) except <CHF> and <EUR> permitted payment type 5: only <EUR> permitted	CHF
O	+IBAN Details	ibanDetails					Data for a domestic or foreign IBAN-transaction (SEPA included)	
O	++SEPA Indicator	sepaIndicator	X				Indicates, whether the payment is an SEPA payment. Must always be <true>	true
M	+Creditor Account	creditorAccount						
M	++Type	type		X		4	Allowed account identification type for the creditor account. The creditor account must always be an IBAN.	IBAN
M	++Identification	identification		X		34	Contains the creditors (payees) account number as an IBAN.	DE12500105170648489890
O	+Creditor Agent							
M {or}	++BIC	bic		X		11	BIC of the creditor's financial institution. If used, then "Clearing System Member Identification" must not be used.	UBSWCHZH80A
M {or}	++Clearing System Member Identification	clearingSystemMemberIdentification					Information used to identify a member within a clearing system.	
M	++++Code	code		X		5	Note: Only "CHBCC" is permitted in Switzerland.	CHBCC
M	++++Member Identification	memberId		X		35	Identification of a member of a clearing system. Clearing ID (Bank Code, "National Identifier") of the receiver institution.	99999
O	++Charge Bearer	chargeBearer		X		4	Permitted codes are: • DEBT Born by Debtor (ex OUR) • CRED Born by Creditor (ex BEN) • SHAR Shared (ex. SHA) • SLEV Service Level	SLEV
M	++Creditor	creditor						
M	++Name	name		X		140	Contains the creditors (payees) name.	Robert Schneider SA
M	++Postal Address	postalAddress					Contains the creditors (payees) postal address.	
M {or}	++Structured	structured					Structured address.	
M	++++Street Name	streetName	X			70	In case of a structured address, this field contains the street name.	Rosenauweg
O	++++Building Number	buildingNumber	X			16	In case of a structured address, this field contains the building number.	4
M	++++Post Code	postCode	X			16	In case of a structured address, this field contains the post code.	80036
M	++++Town Name	townName	X			35	In case of a structured address, this field contains the town name.	Muenchen
M	++++Country	country	X			2	In case of a structured address, this field contains the country.	DE
M {or}	++Unstructured	unstructured					Unstructured address	
M	++++Address Lines	addressLines	X			140	Only two address lines allowed. (2x70)	Rosenauweg 4, 80036 Muenchen
M	++++Country	country	X			2	In case address lines are used, this field must contain the country code.	DE
O	+Remittance Information	remittanceInformation	X			140	Unstructured message	Invoice 2317 due April 1st 2018
O	+Remittance Reference	remittanceReference					Structured reference.	
M	++Type	type		X		4	Reference Type . Permitted codes are: • QR • SCOR	SCOR
M	++Reference	reference	X			27	Contains the Creditor Reference (ISO 11649) or QR-Reference	RF712348231
O	+ISR Details	isrDetails					Details for an ISR transaction	
M	+Creditor Account	creditorAccount						
M	++Type	type		X		5	Allowed account identification type for the creditor account. The creditor account must always be an OTHER.	OTHER
M	++Identification	identification		X		34	Contains the creditors (payees) ISR-Participant number	01-39139-1
M	+Creditor	creditor						
M	++Name	name		X		140	Contains the creditors (payees) name.	
M	++Postal Address	postalAddress	X				Contains the creditors (payees) postal address.	
M {or}	++Structured	structured		X			For sub-elements see element IBAN Details.	
M {or}	++Unstructured	unstructured		X			For sub-elements see element IBAN Details.	
O	+Remittance Reference	remittanceReference						
M	++Type	type		X		4	Reference type . Permitted code: ESR	ESR
M	++Reference	reference	X			27	Contains the ISR-Reference	210000000003139471430009017
O	+Other Details	otherDetails					Details for a domestic or foreign transaction / proprietary account.	
M	+Creditor Account	creditorAccount						
M	++Type	type		X		5	Allowed account identification type for the creditor account. The creditor account must always be OTHER.	OTHER
M	++Identification	identification		X		34	Contains the creditors (payees), IBAN or bank account number.	4124-14345-23424
O {or}	++Creditor Agent	creditorAgent						
M {or}	++BIC	bic				11	BIC of the creditor's financial institution. If used, then "Clearing System Member Identification" must not be used.	UBSWCHZH80A
O {or}	++Clearing System Member Identification	clearingSystemMemberIdentification					Information used to identify a member within a clearing system.	
M	++++Code	code		X		5	Note: Only "CHBCC" is permitted in Switzerland.	CHBCC
M	++++Member Identification	memberId		X		35	Identification of a member of a clearing system. Clearing ID (Bank Code, "National Identifier") of the receiver institution.	99999
O	++Charge Bearer	chargeBearer		X		4	Permitted codes are: • DEBT Born by Debtor (ex OUR) • CRED Born by Creditor (ex BEN) • SHAR Shared (ex. SHA) • SLEV Service Level	DEBT
M	+Creditor	creditor					For sub-elements see element IBAN Details.	
O	+Remittance Information	remittanceInformation		X		140	Contains an unstructured message.	
O	+Remittance Reference	remittanceReference						
M	++Type	type		X		4	Reference Type	
M	++Reference	reference		X		27	Contains the Creditor Reference (ISO 11649) or any other reference.	

Corporate API - /payments/{submissionId}/status - response							
Status	Message Item	Technical	Length	Description			Example
M	Message Identification	messageId	35	Format [UUID 32l], formatted as IETF's RFC 4122 v4 lowercase, no hyphen]			eb6305c9117f49deae016487c27b42d
M	Status Code	statusCode	4	Code telling the TPP, if a payment call was technically accepted, partially accepted or rejected.			ACCP, PART, RJCT
O	Transactions	transactions		If the payment's final payment status is PART or RJCT, the financial institution will additionally deliver the status for the rejected transactions.			
	+Instruction Identification	instructionId	35	Contains the unique tpp-defined ID for the rejected transaction from the original payment instruction.			tx123456789
M	Status Code	statusCode	4	Code telling the TPP, that the transaction of the payment instruction was rejected.			RJCT
M	+Reason Code	reasonCode	4	Reason code to help with the problem solution.			CURR
O	+Reason Information	reasonInformation	105	The financial institution can optionally add further information in unstructured form in case of a rejection.			currency USD not allowed

4.6. Data Model access to account information (AIS)

4.6.1. Request list of accounts

The requests of this type get authorized via token. No additional attributes are needed.

Corporate API - /accounts - response					
Status	Message Item	Technical	Length	Description	Example
M	Account Identification	accountId	35	Contains the account identification for which information shall be retrieved. The TPP receives this identification from the bank. The IBAN should not be used as account identifier. Format [UUID 32lx, formatted as IETF's RFC 4122 v4 lowercase, no hyphen]	550e8400e29b11d4a716446655440000
M	IBAN	iban	34	Contains the accounts International Banking Account Number (IBAN).	CH9300762011623852957
M	Currency	currency	3	Contains the accounts currency.	CHF
O	Designation	designation	140	Contains the accounts designation/description.	Business Account

4.6.2. Request account information

Corporate API - /accounts/{accountId} - request					
Status	Message Item	Technical	Length	Description	Example
M	Account Identification	accountId	35	Contains the account identification for which information shall be retrieved. The TPP receives this identification from the bank. The IBAN should not be used as account identifier. Format [UUID 32]x, formatted as IETF's RFC 4122 v4 lowercase, no hyphen]	550e8400e29b11d4a716446655440000

Corporate API - /accounts/{accountId} - response					
Status	Message Item	Technical	Length	Description	Example
M	Account Identification	accountId	35	Contains the account identification for which information shall be retrieved. The TPP receives this identification from the bank. The IBAN should not be used as account identifier. Format [UUID 32]x, formatted as IETF's RFC 4122 v4 lowercase, no hyphen]	550e8400e29b11d4a716446655440000
M	IBAN	iban	34	Contains the accounts International Banking Account Number (IBAN).	CH9300762011623852957
M	Currency	currency	3	Contains the accounts currency.	CHF
O	Designation	designation	140	Contains the accounts designation/description.	Business Account

4.6.3. Request account balance

Corporate API - /accounts/{accountId}/balance - request					
Status	Message Item	Technical	Length	Description	Example
M	Account Identification	accountId	tbd	Contains the account identification for which information shall be retrieved. The TPP receives this identification from the bank. The IBAN should not be used as account identifier. Format [UUID 32lx, formatted as IETF's RFC 4122 v4 lowercase, no hyphen]	550e8400e29b11d4a716446655440000
O	Date	date	10	Contains the date for which the balance shall be retrieved. If a past date is delivered, the financial institution always responds with the EOD balance (CLBD). If the requested date = today, the financial institution returns the available balance (ITAV).	2018-03-23

Corporate API - /accounts/{accountId}/balance - response					
Status	Message Item	Technical	Length	Description	Example
M	Date	date	10	Contains the date for which the balance was retrieved.	2018-03-23
M	Balance Type	balanceType	4	Contains a code indicating if the balance returned is booked intraday (ITBD) or booked for a closed day (CLBD).	CLBD, ITBD
M	Balance	balance		Contains the requested account balance.	
M	Currency	currency	3	Contains the currency in which the balance and therefore the account is held.	CHF
M	Amount	amount	18 (12.5)	Contains the requested account balance as amount.	113225.00

4.6.4. Request account transactions

Corporate API - /accounts/{accountId}/transactions - request					
Status	Message Item	Technical	Length	Description	Example
M	Account Identification	accountId	35	Contains the account identification for which information shall be retrieved. The TPP receives this identification from the bank. The IBAN should not be used as account identifier. Format [UUID 32lx, formatted as IETF's RFC 4122 v4 lowercase, no hyphen]	550e8400e29b11d4a716446655440000
O	Date	date	10	Contains the booking date for which the transaction information shall be retrieved.	2018-03-23

Corporate API - /accounts/{accountId}/transactions - response					
Status	Message Item	Technical	Length	Description	Example
M	IBAN	iban	35	Contains the accounts International Banking Account Number (IBAN) of the account owner.	CH9300762011623852957
O	Designation	designation	140	Contains the accounts designation/description.	Firmenkonto
M	Entries	entries	35	Unique bank-defined ID for the entry.	
M	+Entry Identification	entryId	4	Code showing if it is credit or debit entry (CRDT, DBIT)	CRDT
M	+Transaction Type	transactionType	4	Mandatory for business cases ISR and OR-IBAN if. Contains the ISR participant number or QR-IBAN of the account owner.	010001628
O	+Entry Reference	entryReference	27	Additional grouping criteria for business cases ISR and OR-IBAN. Contains the BISR-ID or first 6 characters of the QR-reference	123456
O	+Entry Reference Internal Identification	entryReferenceInternalId	6	Indicator shows if the transaction is a reversing entry. Must always be "true"	TRUE
O	+Reversal Indicator	reversalIndicator	boolean	Indicator shows if the transaction is a reversing entry. Must always be "true"	TRUE
M	+Booking Date	bookingDate	10	Booking date of the transaction. The date is set by the bank.	2018-01-02
M	+Value Date	valueDate	10	Value date of the transaction. The date is set by the bank.	2018-01-02
M	+Amount	amount	18 (12,5)	Amount of entry (in account currency)	
M	++Amount	amount	18 (12,5)	Amount	
M	++Currency	currency	3	Account currency	
O	+Instructed Amount	instructedAmount		Amount in the currency of the instruction	
M	++Amount	amount	18 (12,5)	Amount	6500.25
M	++Source Currency	sourceCurrency	3	Source currency	EUR
M	++Target Currency	targetCurrency	3	Target currency	CHF
O	++Exchange Rate	exchangeRate	11 (0,10)	Exchange Rate	1.13665
O	++Exchange Indicator	exchangeIndicator	4	Indicates whether the amount has to be multiplied or divided by the exchange rate (MULT, DIV). Mandatory if exchange rate is provided.	MULT
O	+Total Charges Amount	totalChargesAmount		Total amount of charges	
M	++Amount	amount	18 (12,5)	Amount	
M	++Currency	currency	3	Currency	
O	+Charge Records	chargeRecords		Details about individual charges	
M	+++Amount	amount	18 (12,5)	Amount	
M	+++Currency	currency	3	Currency	
O	+++Charges Included Indicator	chargesIncludedIndicator	boolean	Indicates whether the charges are included in the entry amount or not.	
O	+++Type	type	35	Type of charge	
M	+Bank Transaction Code	bankTransactionCode		Bank Transaction Code	
M	++Domain Code	domainCode	4	Domain code for the "Bank Transaction Code"	PMNT
M	++Family Code	familyCode	4	Family of the "Bank Transaction Code"	RCDT
M	++Sub-Family Code	subfamilyCode	4	Sub-family code for the "Bank Transaction Code"	DMCT
O	+Additional Entry Information	additionalEntryInformation	140	Further detailed information for Entry	
O	+Transactions	transactions		Booking details for the entry	
M	++Transaction Identification	transactionId	35	Unique bank-defined ID for the transaction on the list.	trx123456789
M	++Transaction Type	transactionType	4	Code showing if the transaction was debited or credited to the account (CRDT, DBIT)	CRDT
O	+End to End Identification	endToEndId	35	Unique reference of the debtor/creditor	ENDTOENDID-001
O	+Bank Transaction Code	bankTransactionCode		Bank Transaction Code	
M	++Domain Code	domainCode	4	Domain code for the "Bank Transaction Code"	PMNT
M	++Family Code	familyCode	4	Family of the "Bank Transaction Code"	RCDT
M	++Sub-Family Code	subfamilyCode	4	Sub-family code for the "Bank Transaction Code"	SALA
M	++Amount	amount	18 (12,5)	Amount of transaction (in account currency)	7388.50
M	++Currency	currency	3	Account currency	CHF
O	+Instructed Amount	instructedAmount		Amount in the currency of the instruction	
M	++Amount	amount	18 (12,5)	Amount	6500.25
M	++Source Currency	sourceCurrency	3	Source currency	EUR
M	++Target Currency	targetCurrency	3	Target currency	CHF
O	++Exchange Rate	exchangeRate	11 (0,10)	Exchange Rate	1.13665
O	++Exchange Indicator	exchangeIndicator	4	Indicates whether the amount has to be multiplied or divided by the exchange rate (MULT, DIV). Mandatory if exchange rate is provided.	MULT
O	+Total Charges Amount	totalChargesAmount		Total amount of charges	
M	++Amount	amount	18 (12,5)	Amount	
M	++Currency	currency	3	Currency	
O	+Charge Records	chargeRecords		Details about individual charges	
M	+++Amount	amount	18 (12,5)	Amount	
M	+++Currency	currency	3	Currency	
O	+++Charges Included Indicator	chargesIncludedIndicator	boolean	Indicates whether the charges are included in the entry amount or not.	
O	+++Type	type	35	Type of charge	
O	+Counterparty	counterparty		Counterparty details (debtor or creditor)	
O	+++Name	name	140	Name of counterparty (debtor name if transaction type = CRDT, creditor name if transaction type = DBIT)	Peter Haller
O	+++Postal Address	postalAddress		Address of counterparty (debtor address if transaction type = CRDT, creditor address if transaction type = DBIT)	
O	+--+Structured			Structured address	
O	+---Street Name	streetName	70		Rosenauweg
O	+---Building Number	buildingNumber	16		4
O	+---Post Code	postCode	16		80036
O	+---Town Name	townName	35		Muenchen
O	+---Country	country	2		DE
O	+---Unstructured			Unstructured address	
O	+----Address Lines	addressLines	140	Only two address lines allowed.	Rosenauweg 4, 80036 Muenchen
O	+----Country	country	2	Entered country must be equal to the field must contain the country.	DE
O	+---Account	account		Account of counterparty (debtor account if transaction type = CRDT, creditor account if transaction type = DBIT)	
M	+---Account Type	type	5	Allowed account identification type (IBAN, OTHER)	IBAN
M	+---Account Identification	identification	35	Account number, IBAN, ISR participant number, bank account number	DE12500105170648489890
O	+---Agent	agent			
M (or	+---BIC	bic	11	BIC of the debtor's financial institution (agent of the counterparty). If used, then "Clearing System Member Identification" must not be used.	UBSWCHZH80A
M (or	+---Clearing System Member Identification	clearingSystemMemberIdentification		Information used to identify a member within a clearing system.	
M	++++Code	code	5	Identification of a clearing system.	CHBCC
M	++++Member Identification	memberId	35	Note: Only "CHBCC" is permitted in Switzerland. Identification of a member of a clearing system. Clearing ID (Bank Code, National Identifier) of the financial institution.	99999
O	+Remittance Information	remittanceInformation	140	Unstructured message.	Invoice 2317 due April 1st 2018
O	+Remittance Reference	remittanceReference		Structured reference	
M	+++Type	type	4	Allowed reference type (QRR, SCOR, ESR)	ESR
M	+++Reference	reference	27	QRR-Reference, Creditor Reference according to ISO 11649 or ISR Reference	210000000003139471430009017
O	+Additional Transaction Information	additionalTransactionInformation	140	Further detailed information for transaction.	

4.7. Mapping to Schemes and Standards

4.7.1. Swiss Corporate API attributes vs. SPS ISO-20022

The attributes used in the specified cases can easily be mapped to SPS ISO-20022 attributes.

Attribute in Corporate API	Attribute in SPS/ISO-20022	Data Type	Length
messageIdentification	MsgId	String	35
initiatingPartyIdentification	InitgPty	String	32
requestedExecution Date	ReqdExctnDt	ISODate	10
debtorAccount	DbtrAcct	String	34
creditorAccount	CdtrAcct	String	34
endToEndId	EndToEndId	String	35
instructedAmount	InstdAmt	Amount	18 (12,5)
currency	Ccy	String	3
amount	Amt	Amount	18 (12,5)
statusCode	TxSts	String	4
reasonCode	StsRsnInf	String	4
reasonInformation	StsRsnInf → Rsn → Prtry	String	35
creditor name	Cdtr → Nm	String	140
postalAddress	PstlAdr	String	140
streetName	StrtNm	String	70
buildingNumber	BldgNb	String	16
postCode	PstCd	String	16
townName	TwnNm	String	35
country	Ctry	String	2
type	RmtInf → Strd → CdtrRefInf → Tp	String	35
reference	RmtInf → Strd → CdtrRefInf → Ref	String	35
remittanceInformation	RmtInf → Ustrd	String	140
bic	CdtrAgt → FinInstnId → BIC	String	10
chargeBearer	SvcLvl → Cd	String	4
designation	n.a.	String	140

transactionType	CdtDbtInd	String	4
dateTime	CreDtTm	ISODateTime	25
bookingDate	BookgDt	ISODate	10
valueDate	ValDt	ISODate	10
balance	Bal	Amount	12,5
balanceType	Bal → Tp	String	4
bookingInstruction	DbtrAcct → Tp → Prtry	String	21
instructionId	PmtInfId	String	35
sepaIndicator	InstrPrty → SvcLvl → Cd	boolean	n.a.
code	ClrSysId → Cd	String	5
memberId	ClrSysId → MmbId	String	35
entryId	Ntry → AddtlInflnd → MsgId	String	35
transactionType	CdtDbtInd	String	35
entryReference	NtryRef	String	27
entryReferenceInternalId	n.a	String	6
reversalIndicator	RvslInd	boolean	n.a.
domainCode	Domn → Cd	String	4
familyCode	Fmly → Cd	String	4
subfamilyCode	SubFmlyCd	String	4
additionalEntryInformation	AddtlNtryInf	String	140
transactionId	TxDtls → Refs → PmtInfId	String	35
sourceCurrency	SrcCcy	String	3
targetCurrency	TrgtCcy	String	3
exchangeRate	XchgRate	String	11 (0,10)
exchangeIndicator	n.a.	String	4
chargesIncludedIndicator	ChrgInclInd	boolean	n.a.
chargeRecords → type	Chrgs → Rcrd → Tp	String	35
additionalTransactionInformation	AddtlTxInf	String	140

Chapter 5. Resources

This section is generated from the Swagger API definition file `corapi-api-swagger.yaml` and describes the API endpoints.

The Swagger file contains a lot of additional information including the schema for requests and responses. To view its content, the easiest way is to use the web based editor: <http://editor.swagger.io>.

Please also consider the JSON examples shipped with this documentation.

5.1. Payments

payment initiation submission

Payment Initiation Service - third party payment instruction submission

5.1.1. Submit payment instructions

POST /payments

Parameters

Type	Name	Description	Schema
Header	Authorization required	Bearer followed by a base64 encoded OAuth access token	string
Header	X-CorAPI-Target-ID required	ID that identifies the provider (e.g., a financial institution). (SCOPE: SIX)	string
Header	X-CorAPI-Client-ID required	ID of the client forwarded to the provider. (SCOPE: FI)	string
Header	X-Correlation-ID required	Unique ID (defined by the caller) which will be reflected back in the response.	string
Header	User-Agent required	Name and version of the Client software	string
Header	X-PSU-IP-Address required	IP address of the user initiating the operation	string
Header	X-PSU-User-Agent required	User agent of the user initiating the operation	string
Body	body required	Payment instruction details as defined by data model.	paymentSubmission Request

Responses

HTTP Code	Description	Schema
201	<p>Created.</p> <p>Headers :</p> <p><code>Location</code> (string) : Location of the submitted payment submission message. Add /status for the status report.</p> <p><code>X-Correlation-ID</code> (string) : Reflects the ID (set by the caller) from the request.</p>	No Content
400	<p>Bad Request - The format of the request was invalid.</p> <p>Headers :</p> <p><code>Content-Type</code> (string) : application/problem+json according to RFC7807.</p> <p><code>Content-Encoding</code> (string) : always UTF-8.</p> <p><code>Content-Language</code> (string) : always en.</p> <p><code>X-Correlation-ID</code> (string) : Reflected client defined ID from request.</p>	commonErrorResponse
401	<p>Unauthorized - Either no token or an invalid (e.g., token expired) token was received.</p> <p>Headers :</p> <p><code>Content-Type</code> (string) : application/problem+json according to RFC7807.</p> <p><code>Content-Encoding</code> (string) : always UTF-8.</p> <p><code>Content-Language</code> (string) : always en.</p> <p><code>X-Correlation-ID</code> (string) : Reflected client defined ID from request.</p>	commonErrorResponse
403	<p>Forbidden - A valid OAuth Token was received, but access was denied.</p> <p>Headers :</p> <p><code>Content-Type</code> (string) : application/problem+json according to RFC7807.</p> <p><code>Content-Encoding</code> (string) : always UTF-8.</p> <p><code>Content-Language</code> (string) : always en.</p> <p><code>X-Correlation-ID</code> (string) : Reflected client defined ID from request.</p>	commonErrorResponse
404	<p>Not Found - Either the endpoint does not exist or a requested resource is not yet available (e.g., account statements)</p> <p>Headers :</p> <p><code>Content-Type</code> (string) : application/problem+json according to RFC7807.</p> <p><code>Content-Encoding</code> (string) : always UTF-8.</p> <p><code>Content-Language</code> (string) : always en.</p> <p><code>X-Correlation-ID</code> (string) : Reflected client defined ID from request.</p>	commonErrorResponse
405	<p>Method Not Allowed</p> <p>Headers :</p> <p><code>Content-Type</code> (string) : application/problem+json according to RFC7807.</p> <p><code>Content-Encoding</code> (string) : always UTF-8.</p> <p><code>Content-Language</code> (string) : always en.</p> <p><code>X-Correlation-ID</code> (string) : Reflected client defined ID from request.</p>	commonErrorResponse

HTTP Code	Description	Schema
500	<p>Internal Server Error</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse

Produces

- application/json

5.1.2. Retrieve a payment submission

```
GET /payments/{submissionId}
```

Parameters

Type	Name	Description	Schema
Header	Authorization required	Bearer followed by a base64 encoded OAuth access token	string
Path	submissionId required	id of payment submission	string
Header	X-CorAPI-Target-ID required	ID that identifies the provider (e.g., a financial institution). (SCOPE: SIX)	string
Header	X-CorAPI-Client-ID required	ID of the client forwarded to the provider. (SCOPE: FI)	string
Header	X-Correlation-ID required	Unique ID (defined by the caller) which will be reflected back in the response.	string
Header	User-Agent required	Name and version of the Client software	string

Responses

HTTP Code	Description	Schema
200	<p>The original payment submission</p> <p>Headers :</p> <p>X-Correlation-ID (string) : Reflects the ID (set by the caller) from the request.</p>	paymentSubmissionOnRequest

HTTP Code	Description	Schema
400	<p>Bad Request - The format of the request was invalid.</p> <p>Headers :</p> <p><code>Content-Type</code> (string) : application/problem+json according to RFC7807.</p> <p><code>Content-Encoding</code> (string) : always UTF-8.</p> <p><code>Content-Language</code> (string) : always en.</p> <p><code>X-Correlation-ID</code> (string) : Reflected client defined ID from request.</p>	commonErrorResponse
401	<p>Unauthorized - Either no token or an invalid (e.g., token expired) token was received.</p> <p>Headers :</p> <p><code>Content-Type</code> (string) : application/problem+json according to RFC7807.</p> <p><code>Content-Encoding</code> (string) : always UTF-8.</p> <p><code>Content-Language</code> (string) : always en.</p> <p><code>X-Correlation-ID</code> (string) : Reflected client defined ID from request.</p>	commonErrorResponse
403	<p>Forbidden - A valid OAuth Token was received, but access was denied.</p> <p>Headers :</p> <p><code>Content-Type</code> (string) : application/problem+json according to RFC7807.</p> <p><code>Content-Encoding</code> (string) : always UTF-8.</p> <p><code>Content-Language</code> (string) : always en.</p> <p><code>X-Correlation-ID</code> (string) : Reflected client defined ID from request.</p>	commonErrorResponse
404	<p>Not Found - Either the endpoint does not exist or a requested resource is not yet available (e.g., account statements)</p> <p>Headers :</p> <p><code>Content-Type</code> (string) : application/problem+json according to RFC7807.</p> <p><code>Content-Encoding</code> (string) : always UTF-8.</p> <p><code>Content-Language</code> (string) : always en.</p> <p><code>X-Correlation-ID</code> (string) : Reflected client defined ID from request.</p>	commonErrorResponse
405	<p>Method Not Allowed</p> <p>Headers :</p> <p><code>Content-Type</code> (string) : application/problem+json according to RFC7807.</p> <p><code>Content-Encoding</code> (string) : always UTF-8.</p> <p><code>Content-Language</code> (string) : always en.</p> <p><code>X-Correlation-ID</code> (string) : Reflected client defined ID from request.</p>	commonErrorResponse
500	<p>Internal Server Error</p> <p>Headers :</p> <p><code>Content-Type</code> (string) : application/problem+json according to RFC7807.</p> <p><code>Content-Encoding</code> (string) : always UTF-8.</p> <p><code>Content-Language</code> (string) : always en.</p> <p><code>X-Correlation-ID</code> (string) : Reflected client defined ID from request.</p>	commonErrorResponse

Produces

- application/json

5.1.3. Retrieve payment submission status information

```
GET /payments/{submissionId}/status
```

Parameters

Type	Name	Description	Schema
Header	Authorization required	Bearer followed by a base64 encoded OAuth access token	string
Path	submissionId required	id of payment submission	string
Header	X-CorAPI-Target-ID required	ID that identifies the provider (e.g., a financial institution). (SCOPE: SIX)	string
Header	X-CorAPI-Client-ID required	ID of the client forwarded to the provider. (SCOPE: FI)	string
Header	X-Correlation-ID required	Unique ID (defined by the caller) which will be reflected back in the response.	string
Header	User-Agent required	Name and version of the Client software	string

Responses

HTTP Code	Description	Schema
200	Status information for the payment submission Headers : X-Correlation-ID (string) : Reflects the ID (set by the caller) from the request.	paymentSubmissionStatus
400	Bad Request - The format of the request was invalid. Headers : Content-Type (string) : application/problem+json according to RFC7807. Content-Encoding (string) : always UTF-8. Content-Language (string) : always en. X-Correlation-ID (string) : Reflected client defined ID from request.	commonErrorResponse

HTTP Code	Description	Schema
401	<p>Unauthorized - Either no token or an invalid (e.g., token expired) token was received.</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
403	<p>Forbidden - A valid OAuth Token was received, but access was denied.</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
404	<p>Not Found - Either the endpoint does not exist or a requested resource is not yet available (e.g., account statements)</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
405	<p>Method Not Allowed</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
500	<p>Internal Server Error</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse

Produces

- application/json

5.2. Accounts

account information

Account Information Service - third party access to bank account data

5.2.1. Retrieve list of authorized accounts

GET /accounts

Parameters

Type	Name	Description	Schema
Header	Authorization required	Bearer followed by a base64 encoded OAuth access token	string
Header	X-CorAPI-Target-ID required	ID that identifies the provider (e.g., a financial institution). (SCOPE: SIX)	string
Header	X-CorAPI-Client-ID required	ID of the client forwarded to the provider. (SCOPE: FI)	string
Header	X-Correlation-ID required	Unique ID (defined by the caller) which will be reflected back in the response.	string
Header	User-Agent required	Name and version of the Client software	string

Responses

HTTP Code	Description	Schema
200	Returns a list of authorized accounts. Headers : X-Correlation-ID (string) : Reflects the ID (set by the caller) from the request.	< accountItem > array
400	Bad Request - The format of the request was invalid. Headers : Content-Type (string) : application/problem+json according to RFC7807. Content-Encoding (string) : always UTF-8. Content-Language (string) : always en. X-Correlation-ID (string) : Reflected client defined ID from request.	commonErrorResponse
401	Unauthorized - Either no token or an invalid (e.g., token expired) token was received. Headers : Content-Type (string) : application/problem+json according to RFC7807. Content-Encoding (string) : always UTF-8. Content-Language (string) : always en. X-Correlation-ID (string) : Reflected client defined ID from request.	commonErrorResponse

HTTP Code	Description	Schema
403	<p>Forbidden - A valid OAuth Token was received, but access was denied.</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
404	<p>Not Found - Either the endpoint does not exist or a requested resource is not yet available (e.g., account statements)</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
405	<p>Method Not Allowed</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
500	<p>Internal Server Error</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse

Examples

Response

Example for a list of accounts:

```
[
  {
    "id" : "550e8400e29b11d4a716446655440000",
    "account" : {
      "type" : "IBAN",
      "identification" : "CH9300762011623852957"
    },
    "currency" : "CHF",
    "designation" : "Firmenkonto",
    "_links" : {
      "self" : "/accounts/550e8400e29b11d4a716446655440000",
      "balance" : "/accounts/550e8400e29b11d4a716446655440000/balance",
      "transactions" : "/accounts/550e8400e29b11d4a716446655440000/transactions"
    }
  }
]
```

Produces

- application/json

5.2.2. Retrieve information about a single specific account

GET /accounts/{accountId}

Parameters

Type	Name	Description	Schema
Header	Authorization required	Bearer followed by a base64 encoded OAuth access token	string
Path	accountId required	id of account	string
Header	X-CorAPI-Target-ID required	ID that identifies the provider (e.g., a financial institution). (SCOPE: SIX)	string
Header	X-CorAPI-Client-ID required	ID of the client forwarded to the provider. (SCOPE: FI)	string
Header	X-Correlation-ID required	Unique ID (defined by the caller) which will be reflected back in the response.	string
Header	User-Agent required	Name and version of the Client software	string

Responses

HTTP Code	Description	Schema
200	<p>Returns the details of an account.</p> <p>Headers :</p> <p>X-Correlation-ID (string) : Reflects the ID (set by the caller) from the request.</p>	accountItem
400	<p>Bad Request - The format of the request was invalid.</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
401	<p>Unauthorized - Either no token or an invalid (e.g., token expired) token was received.</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
403	<p>Forbidden - A valid OAuth Token was received, but access was denied.</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
404	<p>Not Found - Either the endpoint does not exist or a requested resource is not yet available (e.g., account statements)</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
405	<p>Method Not Allowed</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse

HTTP Code	Description	Schema
500	<p>Internal Server Error</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse

Examples

Response

Example for account details:

```
{
  "id" : "550e8400e29b11d4a716446655440000",
  "account" : {
    "type" : "IBAN",
    "identification" : "CH9300762011623852957"
  },
  "currency" : "CHF",
  "designation" : "Firmenkonto",
  "_links" : {
    "self" : "/accounts/550e8400e29b11d4a716446655440000",
    "balance" : "/accounts/550e8400e29b11d4a716446655440000/balance",
    "transactions" : "/accounts/550e8400e29b11d4a716446655440000/transactions"
  }
}
```

Produces

- application/json

5.2.3. Retrieve account balance information

```
GET /accounts/{accountId}/balance
```

Description

Retrieve account balance information.

- * Returns the intraday booked balance (ITBD), if called without a date.
- * Returns the intraday booked balance (ITBD), if called for the current date.
- * Returns the closing booked balance (CLBD), if called for a past date.

Parameters

Type	Name	Description	Schema	Default
Header	Authorization <i>required</i>	Bearer followed by a base64 encoded OAuth access token	string	
Path	accountId <i>required</i>	id of account	string	
Query	date <i>optional</i>	The date to query, formatted as yyyy-mm-dd.	string(date)	"the current date"
Header	X-CorAPI-Target-ID <i>required</i>	ID that identifies the provider (e.g., a financial institution). (SCOPE: SIX)	string	
Header	X-CorAPI-Client-ID <i>required</i>	ID of the client forwarded to the provider. (SCOPE: FI)	string	
Header	X-Correlation-ID <i>required</i>	Unique ID (defined by the caller) which will be reflected back in the response.	string	
Header	User-Agent <i>required</i>	Name and version of the Client software	string	

Responses

HTTP Code	Description	Schema
200	Returns account balance information. Headers : <i>X-Correlation-ID</i> (string) : Reflects the ID (set by the caller) from the request.	accountBalanceItem
400	Bad Request - The format of the request was invalid. Headers : <i>Content-Type</i> (string) : application/problem+json according to RFC7807. <i>Content-Encoding</i> (string) : always UTF-8. <i>Content-Language</i> (string) : always en. <i>X-Correlation-ID</i> (string) : Reflected client defined ID from request.	commonErrorResponse
401	Unauthorized - Either no token or an invalid (e.g., token expired) token was received. Headers : <i>Content-Type</i> (string) : application/problem+json according to RFC7807. <i>Content-Encoding</i> (string) : always UTF-8. <i>Content-Language</i> (string) : always en. <i>X-Correlation-ID</i> (string) : Reflected client defined ID from request.	commonErrorResponse

HTTP Code	Description	Schema
403	<p>Forbidden - A valid OAuth Token was received, but access was denied.</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
404	<p>Not Found - Either the endpoint does not exist or a requested resource is not yet available (e.g., account statements)</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
405	<p>Method Not Allowed</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
500	<p>Internal Server Error</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse

Examples

Response

Example for CLBD balance:

```
{
  "date" : "2018-02-15",
  "balanceType" : "CLBD",
  "balance" : {
    "currency" : "CHF",
    "amount" : "9876.75"
  },
  "_links" : {
    "self" : "/accounts/550e8400e29b11d4a716446655440000/balance",
    "account" : "/accounts/550e8400e29b11d4a716446655440000",
    "transactions" : "/accounts/550e8400e29b11d4a716446655440000/transactions"
  }
}
```

Example for ITBD balance:

```
{
  "date" : "2018-02-05",
  "balanceType" : "ITBD",
  "balance" : {
    "currency" : "CHF",
    "amount" : "3948.75"
  },
  "_links" : {
    "self" : "/accounts/550e8400e29b11d4a716446655440000/balance",
    "account" : "/accounts/550e8400e29b11d4a716446655440000",
    "transactions" : "/accounts/550e8400e29b11d4a716446655440000/transactions"
  }
}
```

Produces

- *application/json*

5.2.4. Retrieve transactions of a specific account

```
GET /accounts/{accountId}/transactions
```

Description

- Returns the transaction list of the current day, if called without a date. * Returns the transaction list for a specific day, if called for a past date. In case the specified day has not yet been finalized, the response will be a 404 error.

Parameters

Type	Name	Description	Schema	Default
Header	Authorization <i>required</i>	Bearer followed by a base64 encoded OAuth access token	string	
Path	accountId <i>required</i>	id of account	string	
Query	date <i>optional</i>	The date to query, formatted as yyyy-mm-dd.	string(date)	"the current date"
Header	X-CorAPI-Target-ID <i>required</i>	ID that identifies the provider (e.g., a financial institution). (SCOPE: SIX)	string	
Header	X-CorAPI-Client-ID <i>required</i>	ID of the client forwarded to the provider. (SCOPE: FI)	string	
Header	X-Correlation-ID <i>required</i>	Unique ID (defined by the caller) which will be reflected back in the response.	string	
Header	User-Agent <i>required</i>	Name and version of the Client software	string	

Responses

HTTP Code	Description	Schema
200	Returns transactions of an account. Headers : X-Correlation-ID (string) : Reflects the ID (set by the caller) from the request.	accountTransactionReport
400	Bad Request - The format of the request was invalid. Headers : Content-Type (string) : application/problem+json according to RFC7807. Content-Encoding (string) : always UTF-8. Content-Language (string) : always en. X-Correlation-ID (string) : Reflected client defined ID from request.	commonErrorResponse
401	Unauthorized - Either no token or an invalid (e.g., token expired) token was received. Headers : Content-Type (string) : application/problem+json according to RFC7807. Content-Encoding (string) : always UTF-8. Content-Language (string) : always en. X-Correlation-ID (string) : Reflected client defined ID from request.	commonErrorResponse

HTTP Code	Description	Schema
403	<p>Forbidden - A valid OAuth Token was received, but access was denied.</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
404	<p>Not Found - Either the endpoint does not exist or a requested resource is not yet available (e.g., account statements)</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
405	<p>Method Not Allowed</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
500	<p>Internal Server Error</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse

Examples

Response

Example for a transaction list:

```
"familyCode": "RCDT",
"subFamilyCode": "DMCT"
},
"amount": {
  "currency": "CHF",
  "amount": "250"
},
"additionalEntryInformation": "SAMMELGUTSCHRIFT VOM 29.10.2018"
},
{
  "entryId": "ENTRY123456789",
  "transactionType": "CRDT",
  "bookingDate": "2018-10-29",
  "valueDate": "2018-10-29",
  "bankTransactionCode": {
    "domainCode": "PMNT",
    "familyCode": "RCDT",
    "subFamilyCode": "DMCT"
  },
  "amount": {
    "currency": "CHF",
    "amount": "250"
  },
  "transactions": [
    {
      "transactionId": "trx123456789-01",
      "transactionType": "CRDT",
      "endToEndId": "endToEndId-01",
      "bankTransactionCode": {
        "domainCode": "PMNT",
        "familyCode": "RCDT",
        "subFamilyCode": "DMCT"
      },
      "amount": {
        "currency": "CHF",
        "amount": "250"
      }
    },
    "counterparty": {
      "name": "Barbara Muster",
      "postalAddress": {
        "structured": {
          "streetName": "Rosenauweg",
          "buildingNumber": "4",
          "postCode": "8001",
          "townName": "Zuerich",
          "country": "CH"
        }
      }
    },
    "account": {
      "type": "IBAN",
      "identification": "CH85002582584X1234560"
    }
  ]
}
```

```

        },
        "agent" : {
            "clearingSystemMemberIdentification" : {
                "code" : "CHBCC",
                "memberId" : "99999"
            }
        }
    },
    "remittanceInformation" : "RECHNUNG 67890"
}
]
},
{
    "entryId" : "ENTRY7777777",
    "transactionType" : "CRDT",
    "bookingDate" : "2018-10-29",
    "valueDate" : "2018-10-29",
    "bankTransactionCode" : {
        "domainCode" : "PMNT",
        "familyCode" : "RCDT",
        "subFamilyCode" : "ESCT"
    },
    "amount" : {
        "currency" : "CHF",
        "amount" : "3.47"
    },
    "transactions" : [
        {
            "transactionId" : "trx123456789-01",
            "transactionType" : "CRDT",
            "endToEndId" : "endToEndId-02",
            "bankTransactionCode" : {
                "domainCode" : "PMNT",
                "familyCode" : "RCDT",
                "subFamilyCode" : "ESCT"
            },
            "amount" : {
                "currency" : "CHF",
                "amount" : "3.47"
            },
            "instructedAmount" : {
                "amount" : "3",
                "sourceCurrency" : "EUR",
                "targetCurrency" : "CHF",
                "exchangeRate" : "1.15632286",
                "exchangeIndicator" : "MULT"
            },
            "counterparty" : {
                "name" : "Peter Haller",
                "postalAddress" : {
                    "structured" : {

```

```

        "streetName" : "Rosenauweg",
        "buildingNumber" : "4",
        "postCode" : "80036",
        "townName" : "Muenchen",
        "country" : "DE"
    }
},
"account" : {
    "type" : "IBAN",
    "identification" : "DE12500105170648489890"
}
},
"remittanceInformation" : "RECHNUNG 23456"
}
]
},
{
    "entryId" : "ENTRY66666666",
    "transactionType" : "CRDT",
    "entryReference" : "010026540",
    "bookingDate" : "2018-10-29",
    "valueDate" : "2018-10-29",
    "bankTransactionCode" : {
        "domainCode" : "PMNT",
        "familyCode" : "RCDT",
        "subFamilyCode" : "VCOM"
    },
    "amount" : {
        "currency" : "CHF",
        "amount" : "145.70"
    },
    "transactions" : [
        {
            "transactionId" : "trx123456789-01",
            "transactionType" : "CRDT",
            "endToEndId" : "endToEndId-03",
            "bankTransactionCode" : {
                "domainCode" : "PMNT",
                "familyCode" : "RCDT",
                "subFamilyCode" : "VCOM"
            },
            "amount" : {
                "currency" : "CHF",
                "amount" : "100"
            },
            "counterparty" : {
                "name" : "Max Muster",
                "postalAddress": {
                    "unstructured" : {
                        "addressLines" : [
                            "Bundesplatz 1",

```

```

        "3003 Bern"
    ],
    "country" : "CH"
}
},
"account" : {
    "type" : "IBAN",
    "identification" : "CH85002582584X1234560"
}
},
"remittanceReference" : {
    "type" : "ISR",
    "reference" : "123456789012345678901234567"
}
},
{
    "transactionId" : "trx123456789-01",
    "transactionType" : "CRDT",
    "endToEndId" : "endToEndId-04",
    "bankTransactionCode" : {
        "domainCode" : "PMNT",
        "familyCode" : "RCDT",
        "subFamilyCode" : "VCOM"
    },
    "amount" : {
        "currency" : "CHF",
        "amount" : "45.70"
    },
    "counterparty" : {
        "name" : "Peter Muster",
        "postalAddress": {
            "unstructured" : [
                "addressLines" : [
                    "Bundesplatz 1",
                    "3003 Bern"
                ],
                "country" : "CH"
            }
        },
        "account" : {
            "type" : "IBAN",
            "identification" : "CH85002582584X1234560"
        }
    },
    "remittanceReference" : {
        "type" : "ISR",
        "reference" : "123456789012345678901234567"
    }
}
]
},

```

```
{
    "entryId" : "ENTRY123456789",
    "transactionType" : "DBIT",
    "bookingDate" : "2018-10-29",
    "valueDate" : "2018-10-29",
    "bankTransactionCode" : {
        "domainCode" : "PMNT",
        "familyCode" : "ICDT",
        "subFamilyCode" : "OTHR"
    },
    "amount" : {
        "currency" : "CHF",
        "amount" : "9.57"
    },
    "transactions" : [
        {
            "transactionId" : "trx123456789-01",
            "transactionType" : "DBIT",
            "endToEndId" : "endToEndId-01",
            "bankTransactionCode" : {
                "domainCode" : "PMNT",
                "familyCode" : "ICDT",
                "subFamilyCode" : "OTHR"
            },
            "amount" : {
                "currency" : "CHF",
                "amount" : "9.57"
            },
            "instructedAmount" : {
                "amount" : "10",
                "sourceCurrency" : "USD",
                "targetCurrency" : "CHF",
                "exchangeRate" : "0.957",
                "exchangeIndicator" : "MULT"
            },
            "counterparty" : {
                "name" : "Jan Kowalski",
                "postalAddress" : {
                    "structured" : {
                        "streetName" : "Szczytnicka 9",
                        "postCode" : "50-382",
                        "townName" : "Wroclaw",
                        "country" : "PL"
                    }
                },
                "account" : {
                    "type" : "IBAN",
                    "identification" : "PL7910501575100002345678901"
                },
                "agent" : {
                    "bic" : "INGBPLPW"
                }
            }
        }
    ]
}
```

```

        }
    },
    "remittanceInformation" : "Invoice AB-123-C",
    "additionalTransactionInformation" : "string"
}
]
},
{
    "entryId" : "ENTRY123456789",
    "transactionType" : "DBIT",
    "bookingDate" : "2018-10-29",
    "valueDate" : "2018-10-29",
    "bankTransactionCode" : {
        "domainCode" : "PMNT",
        "familyCode" : "ICDT",
        "subFamilyCode" : "VCOM"
    },
    "amount" : {
        "currency" : "CHF",
        "amount" : "120"
    },
    "additionalEntryInformation" : "ESR VERARBEITUNG VOM 29.10.2018",
    "transactions" : [
        {
            "transactionId" : "trx123456789-01",
            "transactionType" : "DBIT",
            "endToEndId" : "endToEndId-01",
            "bankTransactionCode" : {
                "domainCode" : "PMNT",
                "familyCode" : "ICDT",
                "subFamilyCode" : "VCOM"
            },
            "amount" : {
                "currency" : "CHF",
                "amount" : "120"
            },
            "counterparty" : {
                "name" : "Aero Club der Schweiz",
                "postalAddress": {
                    "unstructured" : {
                        "addressLines" : [
                            "Lidostrasse 5",
                            "6006 Luzern"
                        ],
                        "country" : "CH"
                    }
                },
                "account" : {
                    "type" : "OTHER",
                    "identification" : "01-39139-1"
                }
            }
        }
    ]
}

```

```

        },
        "remittanceReference" : {
            "type" : "ISR",
            "reference" : "047280000701047470007679672"
        }
    }
],
"_links" : {
    "self" : "/accounts/550e8400e29b11d4a716446655440000/transactions",
    "account" : "/accounts/550e8400e29b11d4a716446655440000",
    "balance" : "/accounts/550e8400e29b11d4a716446655440000/balance"
}
}

```

Produces

- application/json

5.3. Iso20022

iso20022 xml

"SPSSTMT": Transport XML account statements & "SPSPMTS": Transport XML payment initiations

5.3.1. Submit an ISO20022 XML payment instruction (PAIN.001)

POST /iso20022/payments

Description

Submit an ISO20022 XML PAIN.001 payment instruction.

The submitted payment instruction must conform to the XML Schema and Implementation Guidelines defined by Swiss Payment Standards;

see: <https://www.six-interbank-clearing.com/en/home/standardization/iso-payments/customer-bank/implementation-guidelines.html>

Parameters

Type	Name	Description	Schema
Header	Authorization required	Bearer followed by a base64 encoded OAuth access token	string
Header	X-CorAPI-Target-ID required	ID that identifies the provider (e.g., a financial institution). (SCOPE: SIX)	string

Type	Name	Description	Schema
Header	X-CorAPI-Client-ID <i>required</i>	ID of the client forwarded to the provider. (SCOPE: FI)	string
Header	X-Correlation-ID <i>required</i>	Unique ID (defined by the caller) which will be reflected back in the response.	string
Header	User-Agent <i>required</i>	Name and version of the Client software	string
Header	X-PSU-IP-Address <i>required</i>	IP address of the user initiating the operation	string
Header	X-PSU-User-Agent <i>required</i>	User agent of the user initiating the operation	string
Body	body <i>required</i>	Payment instruction details as defined by data model.	string

Responses

HTTP Code	Description	Schema
201	<p>Created.</p> <p>Headers :</p> <p>Location (string) : Location of the submitted ISO20022 XML PAIN.001 message. Add /status for the PAIN.002 status report.</p> <p>X-Correlation-ID (string) : Reflects the ID (set by the caller) from the request.</p>	No Content
400	<p>Bad Request - The format of the request was invalid.</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
401	<p>Unauthorized - Either no token or an invalid (e.g., token expired) token was received.</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse

HTTP Code	Description	Schema
403	<p>Forbidden - A valid OAuth Token was received, but access was denied.</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
404	<p>Not Found - Either the endpoint does not exist or a requested resource is not yet available (e.g., account statements)</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
405	<p>Method Not Allowed</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
500	<p>Internal Server Error</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse

Consumes

- application/xml

5.3.2. Retrieve the submitted ISO20022 XML PAIN.001 message.

```
GET /iso20022/payments/{submissionId}
```

Parameters

Type	Name	Description	Schema
Header	Authorization required	Bearer followed by a base64 encoded OAuth access token	string
Path	submissionId required	id of payment submission	string

Type	Name	Description	Schema
Header	X-CorAPI-Target-ID <i>required</i>	ID that identifies the provider (e.g., a financial institution). (SCOPE: SIX)	string
Header	X-CorAPI-Client-ID <i>required</i>	ID of the client forwarded to the provider. (SCOPE: FI)	string
Header	X-Correlation-ID <i>required</i>	Unique ID (defined by the caller) which will be reflected back in the response.	string
Header	User-Agent <i>required</i>	Name and version of the Client software	string

Responses

HTTP Code	Description	Schema
200	Returns the submitted ISO20022 XML PAIN.001 message Headers : X-Correlation-ID (string) : Reflects the ID (set by the caller) from the request.	file
400	Bad Request - The format of the request was invalid. Headers : Content-Type (string) : application/problem+json according to RFC7807. Content-Encoding (string) : always UTF-8. Content-Language (string) : always en. X-Correlation-ID (string) : Reflected client defined ID from request.	commonErrorResponse
401	Unauthorized - Either no token or an invalid (e.g., token expired) token was received. Headers : Content-Type (string) : application/problem+json according to RFC7807. Content-Encoding (string) : always UTF-8. Content-Language (string) : always en. X-Correlation-ID (string) : Reflected client defined ID from request.	commonErrorResponse
403	Forbidden - A valid OAuth Token was received, but access was denied. Headers : Content-Type (string) : application/problem+json according to RFC7807. Content-Encoding (string) : always UTF-8. Content-Language (string) : always en. X-Correlation-ID (string) : Reflected client defined ID from request.	commonErrorResponse

HTTP Code	Description	Schema
404	<p>Not Found - Either the endpoint does not exist or a requested resource is not yet available (e.g., account statements)</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
405	<p>Method Not Allowed</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
500	<p>Internal Server Error</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse

Produces

- application/xml

5.3.3. Retrieve ISO20022 XML status report (PAIN.002) of a payment instruction

```
GET /iso20022/payments/{submissionId}/status
```

Description

Retrieve the XML ISO20022 pain.002 status report.

The returned status report must conform to the XML Schema and Implementation Guidelines defined by Swiss Payment Standards;

see: <https://www.six-interbank-clearing.com/en/home/standardization/iso-payments/customer-bank/implementation-guidelines.html>

Parameters

Type	Name	Description	Schema
Header	Authorization <i>required</i>	Bearer followed by a base64 encoded OAuth access token	string

Type	Name	Description	Schema
Path	submissionId <i>required</i>	id of payment submission	string
Header	X-CorAPI-Target-ID <i>required</i>	ID that identifies the provider (e.g., a financial institution). (SCOPE: SIX)	string
Header	X-CorAPI-Client-ID <i>required</i>	ID of the client forwarded to the provider. (SCOPE: FI)	string
Header	X-Correlation-ID <i>required</i>	Unique ID (defined by the caller) which will be reflected back in the response.	string
Header	User-Agent <i>required</i>	Name and version of the Client software	string

Responses

HTTP Code	Description	Schema
200	<p>Returns the ISO20022 XML PAIN.002 status report of a payment instruction</p> <p>Headers :</p> <p>X-Correlation-ID (string) : Reflects the ID (set by the caller) from the request.</p>	file
400	<p>Bad Request - The format of the request was invalid.</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
401	<p>Unauthorized - Either no token or an invalid (e.g., token expired) token was received.</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
403	<p>Forbidden - A valid OAuth Token was received, but access was denied.</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse

HTTP Code	Description	Schema
404	<p>Not Found - Either the endpoint does not exist or a requested resource is not yet available (e.g., account statements)</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
405	<p>Method Not Allowed</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
500	<p>Internal Server Error</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse

Produces

- application/xml

5.3.4. Retrieve a list of resource links to account statements (CAMT.053)

```
GET /iso20022/statements
```

Description

Retrieve resources links to available account statements.

The returned account statements must conform to the XML Schema and Implementation Guidelines defined by Swiss Payment Standards;

see: <https://www.six-interbank-clearing.com/en/home/standardization/iso-payments/customer-bank/implementation-guidelines.html>

Parameters

Type	Name	Description	Schema
Header	Authorization <i>required</i>	Bearer followed by a base64 encoded OAuth access token	string

Type	Name	Description	Schema
Header	X-CorAPI-Target-ID <i>required</i>	ID that identifies the provider (e.g., a financial institution). (SCOPE: SIX)	string
Header	X-CorAPI-Client-ID <i>required</i>	ID of the client forwarded to the provider. (SCOPE: FI)	string
Header	X-Correlation-ID <i>required</i>	Unique ID (defined by the caller) which will be reflected back in the response.	string
Header	User-Agent <i>required</i>	Name and version of the Client software	string

Responses

HTTP Code	Description	Schema
200	Returns a collection of references to account statements Headers : X-Correlation-ID (string) : Reflects the ID (set by the caller) from the request.	< iso20022ReportReference > array
400	Bad Request - The format of the request was invalid. Headers : Content-Type (string) : application/problem+json according to RFC7807. Content-Encoding (string) : always UTF-8. Content-Language (string) : always en. X-Correlation-ID (string) : Reflected client defined ID from request.	commonErrorResponse
401	Unauthorized - Either no token or an invalid (e.g., token expired) token was received. Headers : Content-Type (string) : application/problem+json according to RFC7807. Content-Encoding (string) : always UTF-8. Content-Language (string) : always en. X-Correlation-ID (string) : Reflected client defined ID from request.	commonErrorResponse
403	Forbidden - A valid OAuth Token was received, but access was denied. Headers : Content-Type (string) : application/problem+json according to RFC7807. Content-Encoding (string) : always UTF-8. Content-Language (string) : always en. X-Correlation-ID (string) : Reflected client defined ID from request.	commonErrorResponse

HTTP Code	Description	Schema
404	<p>Not Found - Either the endpoint does not exist or a requested resource is not yet available (e.g., account statements)</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
405	<p>Method Not Allowed</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
500	<p>Internal Server Error</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse

Produces

- application/json

5.3.5. Retrieves the designated ISO20022 XML CAMT document.

```
GET /iso20022/statements/{reportId}
```

Parameters

Type	Name	Description	Schema
Header	Authorization required	Bearer followed by a base64 encoded OAuth access token	string
Path	reportId required	id of report	string
Header	X-CorAPI-Target-ID required	ID that identifies the provider (e.g., a financial institution). (SCOPE: SIX)	string
Header	X-CorAPI-Client-ID required	ID of the client forwarded to the provider. (SCOPE: FI)	string

Type	Name	Description	Schema
Header	X-Correlation-ID <i>required</i>	Unique ID (defined by the caller) which will be reflected back in the response.	string
Header	User-Agent <i>required</i>	Name and version of the Client software	string

Responses

HTTP Code	Description	Schema
200	Returns the selected ISO20022 XML CAMT document. Headers : X-Correlation-ID (string) : Reflects the ID (set by the caller) from the request.	file
400	Bad Request - The format of the request was invalid. Headers : Content-Type (string) : application/problem+json according to RFC7807. Content-Encoding (string) : always UTF-8. Content-Language (string) : always en. X-Correlation-ID (string) : Reflected client defined ID from request.	commonErrorResponse
401	Unauthorized - Either no token or an invalid (e.g., token expired) token was received. Headers : Content-Type (string) : application/problem+json according to RFC7807. Content-Encoding (string) : always UTF-8. Content-Language (string) : always en. X-Correlation-ID (string) : Reflected client defined ID from request.	commonErrorResponse
403	Forbidden - A valid OAuth Token was received, but access was denied. Headers : Content-Type (string) : application/problem+json according to RFC7807. Content-Encoding (string) : always UTF-8. Content-Language (string) : always en. X-Correlation-ID (string) : Reflected client defined ID from request.	commonErrorResponse
404	Not Found - Either the endpoint does not exist or a requested resource is not yet available (e.g., account statements) Headers : Content-Type (string) : application/problem+json according to RFC7807. Content-Encoding (string) : always UTF-8. Content-Language (string) : always en. X-Correlation-ID (string) : Reflected client defined ID from request.	commonErrorResponse

HTTP Code	Description	Schema
405	<p>Method Not Allowed</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
500	<p>Internal Server Error</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse

Produces

- application/xml

5.4. Directory

participant information

Information about participating parties. (SCOPE: SIX)

5.4.1. Retrieve a list of all registered clients (e.g., third party providers)

```
GET /directory/clients
```

Description

Returns a list of clients with information regarding software and supported use cases. (SCOPE: SIX)

Parameters

Type	Name	Description	Schema	Default
Query	status <i>optional</i>	status=ACTIVE returns only active clients	enum (ACTIVE, INACTIVE, SUSPENDED, ONBOARDING)	"ACTIVE"
Header	X-Correlation-ID <i>required</i>	Unique ID (defined by the caller) which will be reflected back in the response.	string	
Header	User-Agent <i>required</i>	Name and version of the Client software	string	

Responses

HTTP Code	Description	Schema
200	<p>List of clients</p> <p>Headers :</p> <p>X-Correlation-ID (string) : Reflects the ID (set by the caller) from the request.</p>	< directoryClientIte m > array
400	<p>Bad Request - The format of the request was invalid.</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorRes ponse
401	<p>Unauthorized - Either no token or an invalid (e.g., token expired) token was received.</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorRes ponse
403	<p>Forbidden - A valid OAuth Token was received, but access was denied.</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorRes ponse
404	<p>Not Found - Either the endpoint does not exist or a requested resource is not yet available (e.g., account statements)</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorRes ponse
405	<p>Method Not Allowed</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorRes ponse

HTTP Code	Description	Schema
500	<p>Internal Server Error</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse

Produces

- application/json

5.4.2. Retrieve the information for a registered client (e.g. third party provider)

```
GET /directory/clients/{clientId}
```

Description

Returns information regarding software and supported use cases of a client. (SCOPE: SIX)

Parameters

Type	Name	Description	Schema
Path	clientId <i>required</i>	id of the client	string
Header	X-Correlation-ID <i>required</i>	Unique ID (defined by the caller) which will be reflected back in the response.	string
Header	User-Agent <i>required</i>	Name and version of the Client software	string

Responses

HTTP Code	Description	Schema
200	<p>Information for a client</p> <p>Headers :</p> <p>X-Correlation-ID (string) : Reflects the ID (set by the caller) from the request.</p>	directoryClientItem

HTTP Code	Description	Schema
400	<p>Bad Request - The format of the request was invalid.</p> <p>Headers :</p> <p><code>Content-Type</code> (string) : application/problem+json according to RFC7807.</p> <p><code>Content-Encoding</code> (string) : always UTF-8.</p> <p><code>Content-Language</code> (string) : always en.</p> <p><code>X-Correlation-ID</code> (string) : Reflected client defined ID from request.</p>	commonErrorResponse
401	<p>Unauthorized - Either no token or an invalid (e.g., token expired) token was received.</p> <p>Headers :</p> <p><code>Content-Type</code> (string) : application/problem+json according to RFC7807.</p> <p><code>Content-Encoding</code> (string) : always UTF-8.</p> <p><code>Content-Language</code> (string) : always en.</p> <p><code>X-Correlation-ID</code> (string) : Reflected client defined ID from request.</p>	commonErrorResponse
403	<p>Forbidden - A valid OAuth Token was received, but access was denied.</p> <p>Headers :</p> <p><code>Content-Type</code> (string) : application/problem+json according to RFC7807.</p> <p><code>Content-Encoding</code> (string) : always UTF-8.</p> <p><code>Content-Language</code> (string) : always en.</p> <p><code>X-Correlation-ID</code> (string) : Reflected client defined ID from request.</p>	commonErrorResponse
404	<p>Not Found - Either the endpoint does not exist or a requested resource is not yet available (e.g., account statements)</p> <p>Headers :</p> <p><code>Content-Type</code> (string) : application/problem+json according to RFC7807.</p> <p><code>Content-Encoding</code> (string) : always UTF-8.</p> <p><code>Content-Language</code> (string) : always en.</p> <p><code>X-Correlation-ID</code> (string) : Reflected client defined ID from request.</p>	commonErrorResponse
405	<p>Method Not Allowed</p> <p>Headers :</p> <p><code>Content-Type</code> (string) : application/problem+json according to RFC7807.</p> <p><code>Content-Encoding</code> (string) : always UTF-8.</p> <p><code>Content-Language</code> (string) : always en.</p> <p><code>X-Correlation-ID</code> (string) : Reflected client defined ID from request.</p>	commonErrorResponse
500	<p>Internal Server Error</p> <p>Headers :</p> <p><code>Content-Type</code> (string) : application/problem+json according to RFC7807.</p> <p><code>Content-Encoding</code> (string) : always UTF-8.</p> <p><code>Content-Language</code> (string) : always en.</p> <p><code>X-Correlation-ID</code> (string) : Reflected client defined ID from request.</p>	commonErrorResponse

Produces

- application/json

5.4.3. Retrieve a list of all registered providers (e.g., financial institutions)

```
GET /directory/providers
```

Description

Returns a list of providers with information regarding software and supported use cases. (SCOPE: SIX)

Parameters

Type	Name	Description	Schema	Default
Query	status <i>optional</i>	status=ACTIVE returns only active clients	enum (ACTIVE, INACTIVE, SUSPENDED, ONBOARDING)	"ACTIVE"
Header	X-Correlation-ID <i>required</i>	Unique ID (defined by the caller) which will be reflected back in the response.	string	
Header	User-Agent <i>required</i>	Name and version of the Client software	string	

Responses

HTTP Code	Description	Schema
200	List of providers Headers : X-Correlation-ID (string) : Reflects the ID (set by the caller) from the request.	< directoryProviderItem > array
400	Bad Request - The format of the request was invalid. Headers : Content-Type (string) : application/problem+json according to RFC7807. Content-Encoding (string) : always UTF-8. Content-Language (string) : always en. X-Correlation-ID (string) : Reflected client defined ID from request.	commonErrorResponse

HTTP Code	Description	Schema
401	<p>Unauthorized - Either no token or an invalid (e.g., token expired) token was received.</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
403	<p>Forbidden - A valid OAuth Token was received, but access was denied.</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
404	<p>Not Found - Either the endpoint does not exist or a requested resource is not yet available (e.g., account statements)</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
405	<p>Method Not Allowed</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
500	<p>Internal Server Error</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse

Produces

- application/json

5.4.4. Retrieve the information for a registered provider (e.g., financial institutions)

```
GET /directory/providers/{providerId}
```

Description

Returns information regarding software and supported use cases of a provider. (SCOPE: SIX)

Parameters

Type	Name	Description	Schema
Path	providerId <i>required</i>	id of the provider	string
Header	X-Correlation-ID <i>required</i>	Unique ID (defined by the caller) which will be reflected back in the response.	string
Header	User-Agent <i>required</i>	Name and version of the Client software	string

Responses

HTTP Code	Description	Schema
200	Information for a provider Headers : <i>X-Correlation-ID</i> (string) : Reflects the ID (set by the caller) from the request.	directoryProviderItem
400	Bad Request - The format of the request was invalid. Headers : <i>Content-Type</i> (string) : application/problem+json according to RFC7807. <i>Content-Encoding</i> (string) : always UTF-8. <i>Content-Language</i> (string) : always en. <i>X-Correlation-ID</i> (string) : Reflected client defined ID from request.	commonErrorResponse
401	Unauthorized - Either no token or an invalid (e.g., token expired) token was received. Headers : <i>Content-Type</i> (string) : application/problem+json according to RFC7807. <i>Content-Encoding</i> (string) : always UTF-8. <i>Content-Language</i> (string) : always en. <i>X-Correlation-ID</i> (string) : Reflected client defined ID from request.	commonErrorResponse

HTTP Code	Description	Schema
403	<p>Forbidden - A valid OAuth Token was received, but access was denied.</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
404	<p>Not Found - Either the endpoint does not exist or a requested resource is not yet available (e.g., account statements)</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
405	<p>Method Not Allowed</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
500	<p>Internal Server Error</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse

Produces

- application/json

5.5. Consents

consent information

Information about granted consents.

5.5.1. Retrieve a list of consents

```
GET /consents
```

Description

The consents endpoint allows the client (e.g., a third party provider) to query the consents for a given corporate user. The provider (e.g., a financial institution) must return the consents linked with the access token supplied with the request.

Parameters

Type	Name	Description	Schema
Header	Authorization required	Bearer followed by a base64 encoded OAuth access token	string
Header	X-CorAPI-Target-ID required	ID that identifies the provider (e.g., a financial institution). (SCOPE: SIX)	string
Header	X-CorAPI-Client-ID required	ID of the client forwarded to the provider. (SCOPE: FI)	string
Header	X-Correlation-ID required	Unique ID (defined by the caller) which will be reflected back in the response.	string
Header	User-Agent required	Name and version of the Client software	string

Responses

HTTP Code	Description	Schema
200	List of consents Headers : <i>X-Correlation-ID</i> (string) : Reflects the ID (set by the caller) from the request.	< consentItem > array
400	Bad Request - The format of the request was invalid. Headers : <i>Content-Type</i> (string) : application/problem+json according to RFC7807. <i>Content-Encoding</i> (string) : always UTF-8. <i>Content-Language</i> (string) : always en. <i>X-Correlation-ID</i> (string) : Reflected client defined ID from request.	commonErrorResponse
401	Unauthorized - Either no token or an invalid (e.g., token expired) token was received. Headers : <i>Content-Type</i> (string) : application/problem+json according to RFC7807. <i>Content-Encoding</i> (string) : always UTF-8. <i>Content-Language</i> (string) : always en. <i>X-Correlation-ID</i> (string) : Reflected client defined ID from request.	commonErrorResponse

HTTP Code	Description	Schema
403	<p>Forbidden - A valid OAuth Token was received, but access was denied.</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
404	<p>Not Found - Either the endpoint does not exist or a requested resource is not yet available (e.g., account statements)</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
405	<p>Method Not Allowed</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
500	<p>Internal Server Error</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse

Example Response

```
[
  {
    "useCase" : "ais",
    "accounts" : [
      {
        "accountId": "13579",
        "iban": "CH123131231231231",
        "currency": "CHF",
        "designation": "Kontokorrent ABC"
      },
      {
        "accountId": "24680",
        "iban": "CH0938409809823",
        "currency": "EUR",
        "designation": "Kontokorrent DEF"
      }
    ]
  },
  {
    "useCase" : "pis",
    "accounts" : [
      {
        "accountId": "13579",
        "iban": "CH123131231231231",
        "currency": "CHF",
        "designation": "Kontokorrent ABC",
        "allowedCurrencies": [ "CHF", "EUR", "USD" ]
      },
      {
        "accountId": "24680",
        "iban": "CH0938409809823",
        "currency": "EUR",
        "designation": "Kontokorrent DEF",
        "allowedCurrencies": [ "EUR" ]
      }
    ]
  }
]
```

Produces

- `application/json`

5.6. Oauth

oauth

OAuth endpoint

5.6.1. Get the OAuth access and refresh token

```
POST /oauth/token
```

Description

Returns the OAuth access and refresh token for the specified client and target

Parameters

Type	Name	Description	Schema
FormData	grant_type <i>required</i>	either authorization_code or refresh_token	enum (authorization_code, refresh_token)
FormData	code <i>optional</i>	authorization code if applicable	string
FormData	refresh_token <i>optional</i>	refresh token if applicable	string
FormData	redirect_uri <i>optional</i>	original redirect uri if applicable	string
FormData	client_id <i>required</i>	client id	string
Header	X-CorAPI-Target-ID <i>required</i>	ID that identifies the provider (e.g., a financial institution). (SCOPE: SIX)	string
Header	X-CorAPI-Client-ID <i>required</i>	ID of the client forwarded to the provider. (SCOPE: FI)	string
Header	X-Correlation-ID <i>required</i>	Unique ID (defined by the caller) which will be reflected back in the response.	string
Header	User-Agent <i>required</i>	Name and version of the Client software	string
Header	X-PSU-IP-Address <i>required</i>	IP address of the user initiating the operation	string
Header	X-PSU-User-Agent <i>required</i>	User agent of the user initiating the operation	string

Responses

HTTP Code	Description	Schema
200	<p>OAuth token response</p> <p>Headers :</p> <p>X-Correlation-ID (string) : Reflects the ID (set by the caller) from the request.</p> <p>Cache-Control (string) : must be: no-store. Default : "no-store"</p> <p>Pragma (string) : must be: no-cache. Default : "no-cache"</p>	oauthTokenResponse
400	<p>Bad Request - The format of the request was invalid.</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
401	<p>Unauthorized - Either no token or an invalid (e.g., token expired) token was received.</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
403	<p>Forbidden - A valid OAuth Token was received, but access was denied.</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
404	<p>Not Found - Either the endpoint does not exist or a requested resource is not yet available (e.g., account statements)</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse
405	<p>Method Not Allowed</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse

HTTP Code	Description	Schema
500	<p>Internal Server Error</p> <p>Headers :</p> <p>Content-Type (string) : application/problem+json according to RFC7807.</p> <p>Content-Encoding (string) : always UTF-8.</p> <p>Content-Language (string) : always en.</p> <p>X-Correlation-ID (string) : Reflected client defined ID from request.</p>	commonErrorResponse

Consumes

- application/x-www-form-urlencoded

Produces

- application/json

5.7. Diagnostics

diagnostics

Diagnostics endpoint

5.7.1. Returns all specified request headers and additional diagnostic information

```
GET /healthcheck
```

Parameters

Type	Name	Description	Schema
Header	Authorization <i>optional</i>	Bearer followed by a base64 encoded OAuth access token	string
Header	User-Agent <i>optional</i>	Name and version of the client software	string
Header	X-CorAPI-Target-ID <i>optional</i>	ID of the target, e.g., a financial institution. (SCOPE: SIX)	string
Header	X-CorAPI-Client-ID <i>optional</i>	ID of the client forwarded to the provider. (SCOPE: FI)	string
Header	X-Correlation-ID <i>optional</i>	Unique ID (defined by the caller) which will be reflected back in the response.	string

Type	Name	Description	Schema
Header	X-PSU-IP-Address <i>optional</i>	IP address of the user initiating the operation	string
Header	X-PSU-User-Agent <i>optional</i>	User of the client software	string

Responses

HTTP Code	Description	Schema
200	Health Check Response	healthCheckResponse

Produces

- application/json

5.7.2. Returns all specified request headers, the request body and additional diagnostic information

POST /healthcheck

Parameters

Type	Name	Description	Schema
Header	Authorization <i>optional</i>	Bearer followed by a base64 encoded OAuth access token	string
Header	User-Agent <i>optional</i>	Name and version of the client software	string
Header	X-CorAPI-Target-ID <i>optional</i>	ID of the target, e.g., a financial institution. (SCOPE: SIX)	string
Header	X-CorAPI-Client-ID <i>optional</i>	ID of the client forwarded to the provider. (SCOPE: FI)	string
Header	X-Correlation-ID <i>optional</i>	Unique ID (defined by the caller) which will be reflected back in the response.	string
Header	X-PSU-IP-Address <i>optional</i>	IP address of the user initiating the operation	string

Type	Name	Description	Schema
Header	X-PSU-User-Agent <i>optional</i>	User of the client software	string
Body	body <i>required</i>		string

Responses

HTTP Code	Description	Schema
200	Health Check Response	healthCheckResponse

Produces

- [application/json](#)

Chapter 6. Definitions

This section is generated from the Swagger API definition file `corapi-api-swagger.yaml` and describes the schemas for the api.

The Swagger file contains a lot of additional information including the schema for requests and responses. To view its content, the easiest way is to use the web based editor: <http://editor.swagger.io>.

Please also consider the JSON examples shipped with this documentation.

6.1. paymentSubmissionRequest

Name	Description	Schema
<code>messageId</code> <i>required</i>	Maximal length : 35 Example : "eb6305c91f7f49deaed016487c27b42d"	string
<code>initiatingPart</code> <code>yId</code> <i>required</i>	Maximal length : 32 Example : "TPP01746"	string
<code>requestedExe</code> <code>cutionDate</code> <i>required</i>	Example : "2018-04-07"	string(date)
<code>debtorAccoun</code> <code>t</code> <i>required</i>		paymentIbanAccoun t
<code>bookingInstru</code> <code>ction</code> <i>required</i>		paymentBookingInst ruction
<code>transactions</code> <i>required</i>		< paymentInstructionI tem > array

6.2. paymentInstructionItem

Name	Description	Schema
<code>instructionId</code> <i>required</i>	Maximal length : 35 Example : "INSTR-001"	string
<code>endToEndId</code> <i>required</i>	Maximal length : 35 Example : "ENDTOENDID-001"	string
<code>instructedAm</code> <code>ount</code> <i>required</i>		paymentCurrencyA mount
<code>ibanDetails</code> <i>optional</i>		paymentIBANDetail

Name	Description	Schema
isrDetails <i>optional</i>		paymentISRDetail
otherDetails <i>optional</i>		paymentOtherDetail

6.3. paymentIBANDetail

Name	Description	Schema
sepaIndicator <i>optional</i>	Example : true	boolean
creditorAccount <i>required</i>		paymentIbanAccount
creditorAgent <i>optional</i>		paymentCreditorAgent
creditor <i>required</i>		paymentCreditor
chargeBearer <i>optional</i>		paymentChargeBearerMethod
remittanceReference <i>optional</i>		paymentIbanRemittanceReference
remittanceInformation <i>optional</i>	Example : "Verguetung Juli"	string

6.4. paymentISRDetail

Name	Schema
creditorAccount <i>optional</i>	paymentOtherAccount
creditor <i>optional</i>	paymentCreditor
remittanceReference <i>optional</i>	paymentIsrRemittanceReference

6.5. paymentOtherDetail

Name	Schema
creditorAccount <i>optional</i>	paymentOtherAccount

Name	Schema
creditorAgent <i>optional</i>	paymentCreditorAgent
creditor <i>optional</i>	paymentCreditor
chargeBearer <i>optional</i>	paymentChargeBearerMethod
remittanceReference <i>optional</i>	paymentOtherRemittanceReference

6.6. paymentCreditor

Name	Schema
name <i>required</i>	string
postalAddress <i>required</i>	commonStructuredOrUnstructuredAddress

6.7. paymentChargeBearerMethod

Type : enum (DEBT, CRED, SHAR, SLEV)

6.8. paymentBookingInstruction

Type : enum (SINGLEBOOKING_SIA, SINGLEBOOKING_NOA, BATCHBOOKING_SALA_NOA, BATCHBOOKING_SALA_CND, BATCHBOOKING_NOA, BATCHBOOKING_CND, BATCHBOOKING_CWD)

6.9. paymentSubmissionStatus

Name	Description	Schema
messageId <i>optional</i>	Example : "eb6305c91f7f49deaed016487c27b42d"	string
statusCode <i>optional</i>	Example : "PART"	enum (ACCP, RJCT, PART)
transactions <i>optional</i>		< paymentInstructionItemStatus > array

6.10. paymentInstructionItemStatus

Name	Description	Schema
instructionId <i>optional</i>	Example : "DNCS-20180322-IXN0-TXN0"	string

Name	Description	Schema
statusCode <i>optional</i>	Example : "RJCT"	enum (ACCP, RJCT)
reasonCode <i>optional</i>	Example : "CURR"	string
reasonInformation <i>optional</i>	Example : "currency USD not allowed for payment type ISR"	string

6.11. paymentCreditorAgent

Name	Description	Schema
bic <i>optional</i>	Example : "BDEMMXMM"	string
clearingSystemMemberIdentification <i>optional</i>		commonClearingSystemMemberIdentification

6.12. paymentIbanAccount

Name	Description	Schema
type <i>required</i>	allowed account identification type for the creditor account depends on the payment type. The debtor account must always be an IBAN.	enum (IBAN)
identification <i>required</i>	Maximal length : 34 Pattern : "[A-Z]{2,2}[0-9]{2,2}[a-zA-Z0-9]{1,30}" Example : "CH9300762011623852957"	string

6.13. paymentOtherAccount

Name	Description	Schema
type <i>required</i>	Account identification type. The debtor account must always be an IBAN.	enum (OTHER)
identification <i>required</i>	Maximal length : 34 Example : "762011623852957"	string

6.14. balanceCurrencyAmount

Name	Description	Schema
currency <i>required</i>	Example : "CHF"	string

Name	Description	Schema
amount required	Maximal length : 18 Pattern : "-?[0-9]{1,12}([.][0-9]{1,5})?" Example : "10.25"	string

6.15. paymentCurrencyAmount

Name	Description	Schema
currency required	Example : "CHF"	string
amount required	Maximal length : 18 Pattern : "[0-9]{1,12}([.][0-9]{1,5})?" Example : "10.25"	string

6.16. paymentIbanRemittanceReference

either remittanceReference or remittanceInformation must be set

Name	Description	Schema
type <i>optional</i>		enum (SCOR, QRR)
reference <i>optional</i>	Maximal length : 35 Example : "21000000003139471430009017"	string

6.17. paymentIsrRemittanceReference

either remittanceReference or remittanceInformation must be set

Name	Description	Schema
type <i>optional</i>		enum (ISR)
reference <i>optional</i>	Maximal length : 35 Example : "21000000003139471430009017"	string

6.18. paymentOtherRemittanceReference

either remittanceReference or remittanceInformation must be set

Name	Description	Schema
type <i>optional</i>		enum (SCOR)
reference <i>optional</i>	Maximal length : 35 Example : "21000000003139471430009017"	string

6.19. paymentRemittanceInformation

either remittanceReference or remittanceInformation must be set

Type : string

6.20. paymentEndToEndId

Unique debtor (payee) reference.

Type : string

6.21. accountItem

Name	Description	Schema
id <i>required</i>	Example : "550e8400-e29b-11d4-a716-446655440000"	string
account <i>required</i>		paymentIbanAccount
currency <i>required</i>	Example : "CHF"	string
designation <i>required</i>	Maximal length : 140 Example : "Firmenkonto"	string
_links <i>optional</i>	Contains the paths to additional resources for specific account (e.g. path to transactions resource)	_links

links

Name	Description	Schema
self <i>optional</i>	Example : "/accounts/550e8400-e29b-11d4-a716-446655440000"	string
balance <i>optional</i>	Example : "/accounts/550e8400-e29b-11d4-a716-446655440000/balance"	string
transactions <i>optional</i>	Example : "/accounts/550e8400-e29b-11d4-a716-446655440000/transactions"	string

6.22. accountBalanceItem

Name	Description	Schema
date <i>required</i>	Example : "2010-02-15"	string(date)
balanceType <i>required</i>	Example : "CLBD"	enum (ITBD, CLBD)

Name	Description	Schema
balance <i>required</i>		balanceCurrencyAm ount
_links <i>optional</i>	contains the paths to additional resources for specific account (e.g. path to transactions resource)	_links

[_links](#)

Name	Description	Schema
self <i>optional</i>	Example : "/accounts/550e8400-e29b-11d4-a716-446655440000/balance"	string
account <i>optional</i>	Example : "/accounts/550e8400-e29b-11d4-a716-446655440000"	string
transactions <i>optional</i>	Example : "/accounts/550e8400-e29b-11d4-a716-446655440000/transactions"	string

6.23. accountTransactionReport

Name	Description	Schema
iban <i>required</i>	Example : "CH5481230000001998736"	string
designation <i>optional</i>	Example : "Checking Account"	string
entries <i>required</i>		< accountTransactionE ntry > array
_links <i>optional</i>		_links

[_links](#)

Name	Description	Schema
self <i>optional</i>	Example : "/accounts/550e8400e29b11d4a716446655440000/statements"	string
account <i>optional</i>	Example : "/accounts/550e8400e29b11d4a716446655440000"	string
balance <i>optional</i>	Example : "/accounts/550e8400e29b11d4a716446655440000/balance"	string

6.24. accountTransactionEntry

Name	Description	Schema
entryId <i>required</i>	Example : "ENTRY123456789"	string

Name	Description	Schema
transactionType <i>required</i>		accountTransactionType
entryReference <i>optional</i>	Example : "1.0001628E7"	string
entryReferenceInternalId <i>optional</i>	Example : "010001628"	string
reversalIndicator <i>optional</i>		boolean
bookingDate <i>required</i>	Example : "2018-10-29"	string(date)
valueDate <i>required</i>	Example : "2018-10-29"	string(date)
amount <i>required</i>		paymentCurrencyAmount
instructedAmount <i>optional</i>		accountTransactionInstructedAmount
totalChargesAmount <i>optional</i>		accountTransactionItemCharges
bankTransactionCode <i>required</i>		accountTransactionBankTransactionCode
additionalEntryInformation <i>optional</i>		string
transactions <i>optional</i>		< accountTransactionItem > array

6.25. accountTransactionItem

Name	Description	Schema
transactionId <i>required</i>	Example : "TX12345A987"	string
transactionType <i>required</i>		accountTransactionType
endToEndId <i>optional</i>	Example : "ENDTOENDID-01"	string

Name	Description	Schema
bankTransactionCode <i>optional</i>		accountTransaction BankTransactionCode
amount required		paymentCurrencyAmount
instructedAmount <i>optional</i>		accountTransactionInstructedAmount
totalChargesAmount <i>optional</i>		accountTransactionItemCharges
counterparty <i>optional</i>		accountTransactionCounterparty
remittanceInformation <i>optional</i>		paymentRemittanceInformation
remittanceReference <i>optional</i>		accountTransactionRemittanceReference
additionalTransactionInformation <i>optional</i>		string

6.26. accountTransactionItemCharges

Name	Description	Schema
amount required	Maximal length : 18 Pattern : "[0-9]{1,12}([.][0-9]{1,5})?" Example : "10.25"	string
currency required	Example : "CHF"	string
chargeRecords <i>optional</i>		< accountTransactionItemChargesRecord > array

6.27. accountTransactionItemChargesRecord

Name	Description	Schema
amount optional	Maximal length : 18 Pattern : "[0-9]{1,12}([.][0-9]{1,5})?" Example : "10.25"	string
currency optional	Maximal length : 3 Example : "CHF"	string

Name	Description	Schema
type <i>optional</i>	Maximal length : 35 Example : "Some type of charge"	string
chargesIncludedIndicator <i>optional</i>		boolean

6.28. accountTransactionCounterparty

Name	Description	Schema
name <i>optional</i>	Example : "Hans Muster"	string
postalAddress <i>optional</i>		commonStructuredOrUnstructuredAddresses
account <i>optional</i>		accountTransactionCounterpartyAccount
agent <i>optional</i>		accountTransactionCounterpartyAgent

6.29. accountTransactionCounterpartyAgent

Name	Schema
bic <i>optional</i>	string
clearingSystemMemberIdentification <i>optional</i>	commonClearingSystemMemberIdentification

6.30. accountTransactionInstructedAmount

Name	Description	Schema
amount <i>required</i>	Maximal length : 18 Pattern : "[0-9]{1,12}([.][0-9]{1,5})?" Example : "10.25"	string
sourceCurrency <i>required</i>	Example : "CHF"	string
targetCurrency <i>required</i>	Example : "USD"	string
exchangeRate <i>optional</i>	Example : "0.957"	string

Name	Description	Schema
exchangeIndicator <i>optional</i>	Example : "MULT"	string

6.31. accountTransactionBankTransactionCode

Name	Description	Schema
domainCode <i>required</i>	Example : "PMNT"	string
familyCode <i>required</i>	Example : "RCDT"	string
subFamilyCode <i>required</i>	Example : "DMCT"	string

6.32. accountTransactionTransactionType

Type : enum (CRDT, DBIT)

6.33. accountTransactionCounterpartyAccount

Name	Description	Schema
type <i>required</i>		enum (IBAN, OTHER)
identification <i>required</i>	Maximal length : 34 Example : "CH9300762011623852957"	string

6.34. accountTransactionRemittanceReference

Name	Description	Schema
type <i>optional</i>		enum (SCOR, ISR, QRR)
reference <i>optional</i>	Maximal length : 35 Example : "21000000003139471430009017"	string

6.35. consentItem

Name	Description	Schema
useCase <i>optional</i>	Example : "ais"	string

Name	Description	Schema
accounts <i>optional</i>		< consentAccountItem > array

6.36. consentAccountItem

Name	Description	Schema
accountId <i>required</i>	Example : "13579"	string
iban <i>required</i>	Example : "CH123131231231231"	string
currency <i>optional</i>	Example : "CHF"	string
designation <i>optional</i>	Example : "Savings ABC"	string
allowedCurrencies <i>optional</i>		< string > array

6.37. directoryClientItem

Polymorphism : Composition

Name	Description	Schema
clientId <i>required</i>	Example : "1234"	string
clientConsentBackendUrl <i>required</i>	Example : "https://www.acme.com/consent"	string
clientRedirectIonEndpointUrl <i>required</i>	Example : "https://www.acme.com/start"	string
companyName <i>required</i>	Maximal length : 50 Example : "ACME"	string
contact <i>required</i>		< directoryContact > array
companyLogo <i>required</i>		directoryMultisizeImageReference
marketingImage <i>optional</i>		directoryImageReference

Name	Description	Schema
companyUrl <i>required</i>	Example : "https://www.acme.com"	string
marketingDescription <i>required</i>		directoryMarketingDescription
shortDescription <i>required</i>		directoryShortDescription
status <i>required</i>		directoryParticipantStatus
registrationDate <i>required</i>	Example : "2018-10-11"	string(date)
lastModified <i>required</i>	Example : "2018-11-11T12:39:23+01:00"	string(date-time)
software <i>required</i>		directorySoftwareDescription
useCases <i>optional</i>		< directoryUseCaseItem > array

6.38. directoryProviderItem

Polymorphism : Composition

Name	Description	Schema
providerId <i>required</i>	Example : "1234"	string
providerAuthEndpointUrl <i>required</i>	Example : "https://www.acme.com/oauth/authorize"	string
companyName <i>required</i>	Maximal length : 50 Example : "ACME"	string
contact <i>required</i>		< directoryContact > array
companyLogo <i>required</i>		directoryMultisizeImageReference
marketingImage <i>optional</i>		directoryImageReference
companyUrl <i>required</i>	Example : "https://www.acme.com"	string

Name	Description	Schema
marketingDescription required		directoryMarketingDescription
shortDescription required		directoryShortDescription
status required		directoryParticipantStatus
registrationDate required	Example : "2018-10-11"	string(date)
lastModified required	Example : "2018-11-11T12:39:23+01:00"	string(date-time)
software required		directorySoftwareDescription
useCases optional		< directoryUseCaseItem > array

6.39. directoryParticipantItem

Name	Description	Schema
companyName required	Maximal length : 50 Example : "ACME"	string
contact required		< directoryContact > array
companyLogo required		directoryMultisizeImageReference
marketingImage optional		directoryImageReference
companyUrl required	Example : "https://www.acme.com"	string
marketingDescription required		directoryMarketingDescription
shortDescription required		directoryShortDescription
status required		directoryParticipantStatus

Name	Description	Schema
registrationDate <i>required</i>	Example : "2018-10-11"	string(date)
lastModified <i>required</i>	Example : "2018-11-11T12:39:23+01:00"	string(date-time)
software <i>required</i>		directorySoftwareDescription
useCases <i>optional</i>		< directoryUseCaseItem > array

6.40. directoryContact

Name	Description	Schema
type <i>required</i>		enum (BUSINESS, TECHNICAL)
designation <i>optional</i>	Maximal length : 80	string
emailAddress <i>required</i>	Maximal length : 50	string(email)
phone <i>optional</i>	Maximal length : 20	string

6.41. directorySoftwareDescription

Name	Description	Schema
productName <i>required</i>		directoryProductName
manufacturer <i>required</i>	Maximal length : 50 Example : "ACME Corp."	string
category <i>required</i>		< enum (ACCOUNTING) > array
productInfoUrl <i>required</i>	Example : "https://acme.com/acme_accounting"	string
productLogo <i>required</i>		directoryMultisizeImageReference
marketingImage <i>optional</i>		directoryImageReference

Name	Description	Schema
marketingDescription required		directoryMarketingDescription
shortDescription required		directoryShortDescription

6.42. directoryMarketingDescription

Name	Schema
de required	< string > array
en required	< string > array
fr required	< string > array
it required	< string > array

6.43. directoryShortDescription

Name	Description	Schema
de required	Maximal length : 300	string
en required	Maximal length : 300	string
fr required	Maximal length : 300	string
it required	Maximal length : 300	string

6.44. directoryUseCaseItem

Name	Description	Schema
name required	Example : "ais"	string
version required	Example : "v1"	string
status required	Example : "CERTIFIED"	enum (ONBOARDING, CERTIFIED)

Name	Description	Schema
from required	Example : "2018-10-10T:14:00:00+02:00"	string(date-time)
until optional	Example : "9999-12-31T23:59:59+01:00"	string(date-time)
properties optional		< string, string > map

6.45. directoryProductName

Name	Description	Schema
de required	Maximal length : 50	string
en required	Maximal length : 50	string
fr required	Maximal length : 50	string
it required	Maximal length : 50	string

6.46. directoryMultisizeImageReference

Name	Schema
small required	directoryImageReference
large required	directoryImageReference

6.47. directoryParticipantStatus

Type : enum (ACTIVE, INACTIVE, SUSPENDED, ONBOARDING)

6.48. directoryImageReference

Name	Description	Schema
mimeType required	Example : "image/png"	string
image required	Example : "QUNNRQ=="	string(byte)

6.49. iso20022ReportReference

Name	Schema
name <i>optional</i>	string
description <i>optional</i>	string
type <i>optional</i>	enum (CAMT53, CAMT54)
id <i>optional</i>	string

6.50. oauthTokenResponse

Name	Description	Schema
access_token <i>required</i>		string
token_type <i>required</i>	Default : "bearer"	string
expires_in <i>optional</i>	Minimum value : 1	integer
refresh_token <i>optional</i>		string

6.51. commonClearingSystemMemberIdentification

Name	Description	Schema
code <i>required</i>	Example : "CHBCC"	string
memberId <i>required</i>	Example : "00230"	string

6.52. commonStructuredOrUnstructuredAddress

Name	Schema
structured <i>optional</i>	commonStructuredAddress
unstructured <i>optional</i>	commonUnstructuredAddress

6.53. commonStructuredAddress

Name	Description	Schema
streetName <i>required</i>	Maximal length : 70 Example : "Rue de la gare"	string
buildingNumber <i>optional</i>	Maximal length : 16 Example : "24"	string
postCode <i>required</i>	Maximal length : 16 Example : "2501"	string
townName <i>required</i>	Maximal length : 35 Example : "Biel"	string
country <i>required</i>	Maximal length : 2 Example : "CH"	string

6.54. commonUnstructuredAddress

Name	Description	Schema
addressLines <i>required</i>	max 2 lines of 35 characters. either postalAdress or addressLines must be set.	< string > array
country <i>required</i>	Maximal length : 2 Example : "CH"	string

6.55. commonErrorResponse

Name	Description	Schema
type <i>optional</i>		commonErrorType
title <i>optional</i>	Example : "This is the general problem description"	string
detail <i>optional</i>	Example : "Detailed problem description with respect to the current request, e.g., invalid account number format"	string
instance <i>optional</i>	Example : "path/to/corresponding/resource"	string

6.56. commonErrorType

Error Types for commonErrorResponse. See Appendix for detailed information.

```
Type      : enum      (/problems/INVALID_PAYLOAD,      /problems/MALFORMED_PAYLOAD,
/problems/INVALID_TOKEN,  /problems/EXPIRED_TOKEN,  /problems/INSUFFICIENT_PRIVILEGES,
/problems/NO_ACCESS_TO_RESOURCE,          /problems/RESOURCE_DOES_NOT_EXIST,
/problems/RESOURCE_NOT_READY,           /problems/RESOURCE_TOO_LARGE,
/problems/WRONC_METHOD,                /problems/OPERATION_NOT_ALLOWED,
/problems/TECHNICAL_ERROR)
```

6.57. healthCheckResponse

Name	Description	Schema
requestDateTi me <i>optional</i>	Example : "2018-04-03T07:30:00+01:00"	string(date-time)
receivedHead ers <i>optional</i>		< receivedHeaders > array
receivedPaylo ad <i>optional</i>	Example : "... as received ..."	string

receivedHeaders

Name	Description	Schema
headerName <i>optional</i>	Example : "X-Correlation-ID"	string
headerValue <i>optional</i>	As received	string

Chapter 7. Security and Access Control

7.1. Transport Layer Security

All connections between the corporate API platform, FIs and TPPs are encrypted using TLS 1.2 or higher and strong cipher suites.

7.2. Authentication

All connections are mutually authenticated (2-way TLS):

- The Corporate API authenticates the TPPs using X.509 certificates
- FIs authenticate the Corporate API using X.509 certificates

The list of Certificate Authorities whose certificates are accepted by Corporate API is available on request at SwissCorpAPI@six-group.com

7.3. High Level Authorization

As a result of the certification process, TPPs get authorized for the certified use cases. The platform verifies all requests from TPP's against their certified uses cases. Only admissible requests are forwarded to the FI.

7.4. Consent Management

Consent management authorizes a TPP to access the specific resources of a corporate at a FI. The process to establish such a consent between a corporate user (resource owner) and a bank (resource server) is based on the standard OAuth 2.0 authorization code flow with `response_type=code` as defined in IETF reference [10].

FIs and TPPs must support access (`token_type: bearer`) and refresh tokens in order to successfully execute the authorization code flow.

Deviations from the standard:

- To support initiation of the process in the FI's e-banking application, the process has been extended with an up-front pre-authentication and authorization step.
- OAuth Client-Secret is not supported in the consent management flow: the TPP (client) is authenticated by certificate.
- TPPs must implement a backend endpoint to trigger the standard authorization code flow.
- Token endpoints of FIs are located behind SIX authenticating proxy.
- Corporate API supports exactly one consent per resource owner / FI and TPP.

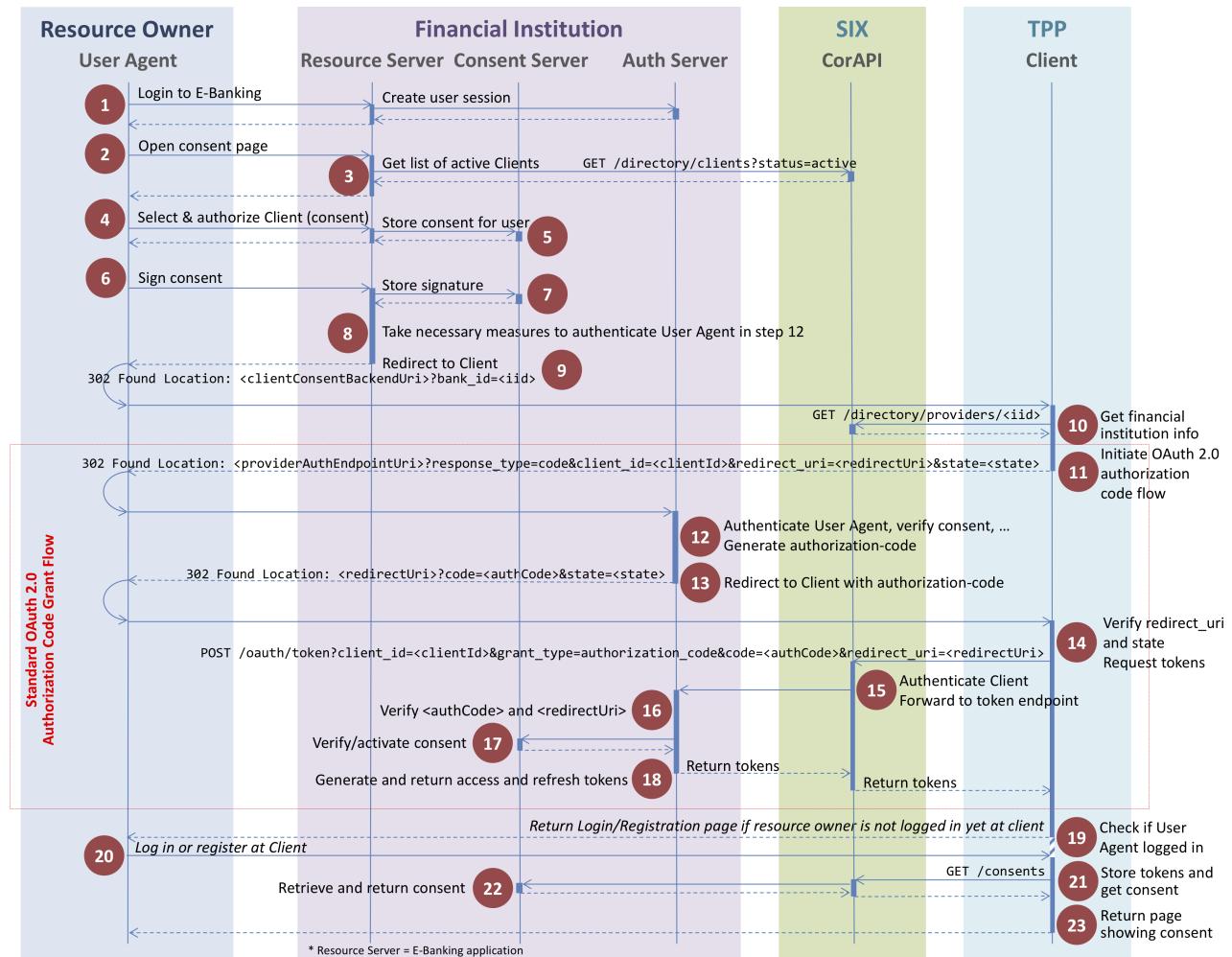


Figure 5. Overview of consent management flow

Consent management flow description

- User logs in to E-Banking application.
- User requests page to give consent.
- E-Banking application retrieves list of active Clients (TPPs) from Corporate API directory.
Example:

```
GET /api/bankingservices/corporate/v1/directory/clients?status=ACTIVE
Host: api-cert.six-group.com
```
- User chooses client and defines consent for access to account and payments and clicks "Save".
- E-Banking application stores consent in user profile.
- Optionally: User has to sign the consent (using 2nd factor).
- Optionally: Store signature (in user profile).
- FI takes measures that allow the authentication of the User Agent during OAuth 2.0 authorization code flow (at step 12). These measures **must be transparent** to the client.
Example: set cookie with one-time-password to be stored in User Agent.
- FI redirects to consent-backend URI of client (URI is part of client info retrieved in step 3). The id of the FI (IID) must be passed to the client as request parameter.

Example:

```
HTTP/1.1 302 Found
Location: https://api.tpp.example.com/corapi/consentLauncher?bank_id=99999
```

10. Consent backend of client receives request and **immediately** retrieves information about the FI from Corporate API directory using the IID from the request.

Example:

```
GET /api/bankingservices/corporate/v1/directory/providers/99999
Host: api-cert.six-group.com
```

11. TPP **immediately** starts standard OAuth 2.0 authorization code flow (`response_type=code`) by redirecting the user agent to the Auth Server of the FI. It is recommended to include a state with the request for preventing cross-site request forgery.

Example:

```
HTTP/1.1 302 Found
Location:
https://secure.fi.example.com/oauth/authorize?response_type=code&client_id=654321&r
edirect_uri=https%3A%2F%2Fapi.tpp.example.com%2Foauth%2Fauthresponse&state=02394202
```

12. Auth Server of FI

- a. authenticates the User Agent using information from step 8,
- b. verifies consent for client with `client_id` from request,
- c. executes any other actions required to verify correctness or request,
- d. if all above ok, it generates an authorization code.

13. If previous step ok the Auth Server redirects to the `redirect_uri` received from client and includes the `authorization_code` and any local state provided by the client (see example below). If the request fails for any reason the Auth Server should inform the resource owner according to OAuth 2.0 specification, 4.1.2.1. [10].

Example:

```
HTTP/1.1 302 Found
Location:
https://api.tpp.example.com/oauth/authresponse?code=abcdefg&state=02394202
```

14. TPP verifies that requested URI matches the `redirect_uri` and that any state sent matches the state returned. If ok, the TPP requests the access and refresh tokens through Corporate API (acts as authenticating proxy towards the FI's token endpoint). For proper routing by Corporate API the request must include a `X-CorAPI-Target-ID` header with the iid of the target FI.

Example from client to SIX:

```

POST /api/bankingservices/corporate/v1/oauth/token
X-CorAPI-Target-ID: 99999
client_id: 654321
grant_type: authorization_code
code: abcdefg
redirect_uri=https://api.tpp.example.com/oauth/authresponse
...
Host: api-cert.six-group.com

```

15. Corporate API token endpoint

- a. authenticates the client
 - b. extracts the IID from the **X-CorAPITarget-ID** request header
 - c. determines the hostname/IP address and basepath of Auth Server token endpoint at target FI
 - d. sends token request with header **X-CorAPI-Client-ID** to target FI
- Example from SIX to FI:

```

POST /proprietaryBackendPath/api/v1/oauth/token
X-CorAPI-Client-ID: 654321
client_id: 654321
grant_type: authorization_code
code: abcdefg
redirect_uri=https://api.tpp.example.com/oauth/authresponse
...
Host: api.fi.example.com

```

16. Auth Server token endpoint of FI

- a. authenticates TPP using request header **X-CorAPI-Client-ID**
- b. verifies authorization code
- c. verifies redirect_uri

17. Consent Server verifies and activates consent

18. Auth Server generates access and refresh tokens and returns them via Corporate API to the client. If the request fails for any reason the Auth Server should inform the resource owner according to OAuth 2.0 specification, 5.2. [10].

19. Client application checks if user is logged in, if not TPP returns login/registration page to user.

20. User logs in to client application or registers.

21. Once the user is logged in the client

- a. stores the IID and the access and refresh tokens in the profile of the user,
 - b. retrieves the consent for the user from Corporate API.
- Example from client to SIX:

```

GET /api/bankingservices/corporate/v1/consents
X-CorAPITarget-ID: 99999
Authorization: Bearer ZX1KMGVYQWlPaUpLVjFRaUxDSmhiR2Np
...
Host: api-cert.six-group.com

```

Example from SIX to FI:

```

GET /proprietaryBackendPath/v1/consents
X-CorAPI-Client-ID: 654321
Authorization: Bearer ZX1KMGVYQWlPaUpLVjFRaUxDSmhiR2Np
...
Host: api.fi.example.com

```

22. Financial Institution

- extracts access token from request
- validates access token
- extracts `client_id` from request header `X-CorAPI-Client-ID`
- verifies that access token has been issued to this `client_id`
- retrieves consent and returns it via Corporate API to client

23. Client stores consent and returns page showing consent to User Agent

Access tokens

Access tokens are issued by the FIs' authorization servers. There are no special requirements regarding content and structure of access tokens. The platform does not validate access tokens.

Rationale:

- Access tokens are treated as blackboxes by Clients and Corporate API.
- No changes required at FI's authorization servers.
- Allows use of symmetric token signing algorithms for JWTs.

Authorization Code and token lifetime

The following table defines mandatory and recommended lifetimes for authorization code and access and refresh tokens.

Code TokenName	Lifetime	Type	Comment
Authorization code	30 sec	Mandatory	The authorization code flow consists of a few redirects without user interaction. It is feasible to keep this lifetime short.
Access token	24 hr	Recommendation	Shorter lifetime could lead to a token refresh flow for almost every resource access.

Code TokenName	Lifetime	Type	Comment
Refresh token	6 months	Recommendation	Consent giving is cumbersome. Refresh token lifetime should be extended with every access token refresh.

Access token expiration must be handled by the token issuer (FI).

Revocation of authorization

Consent can be revoked by the corporate user (resource owner). Access and refresh tokens can be revoked by the issuing FI.

IMPORTANT

Under certain conditions SIX may have to **temporarily** suspend a client or FI from Corporate API. Tokens of suspended Clients shall not be revoked.

FI IAM API endpoints

The FI must expose the authorization endpoint to the internet. The token endpoint may be exposed solely to Corporate API. Corporate API acts as an authenticating proxy towards the token endpoint and protects it from unauthorized access.

7.5. Request validation

Requests from authenticated and authorized Clients must meet the following requirements:

- All mandatory request headers are present and syntactically correct
- All optional request headers are syntactically correct
- No unspecified request headers are present
- JSON or XML payload is syntactically correct

Only valid requests are forwarded to FIs.

7.6. Token refresh

The token refresh flow is implemented according to OAuth 2.0 standard [10].

Example of token expiration and subsequent refresh flow

This example shows a payment initiation submission with an expired access token, subsequent token refresh and resubmission of initial payment initiation.

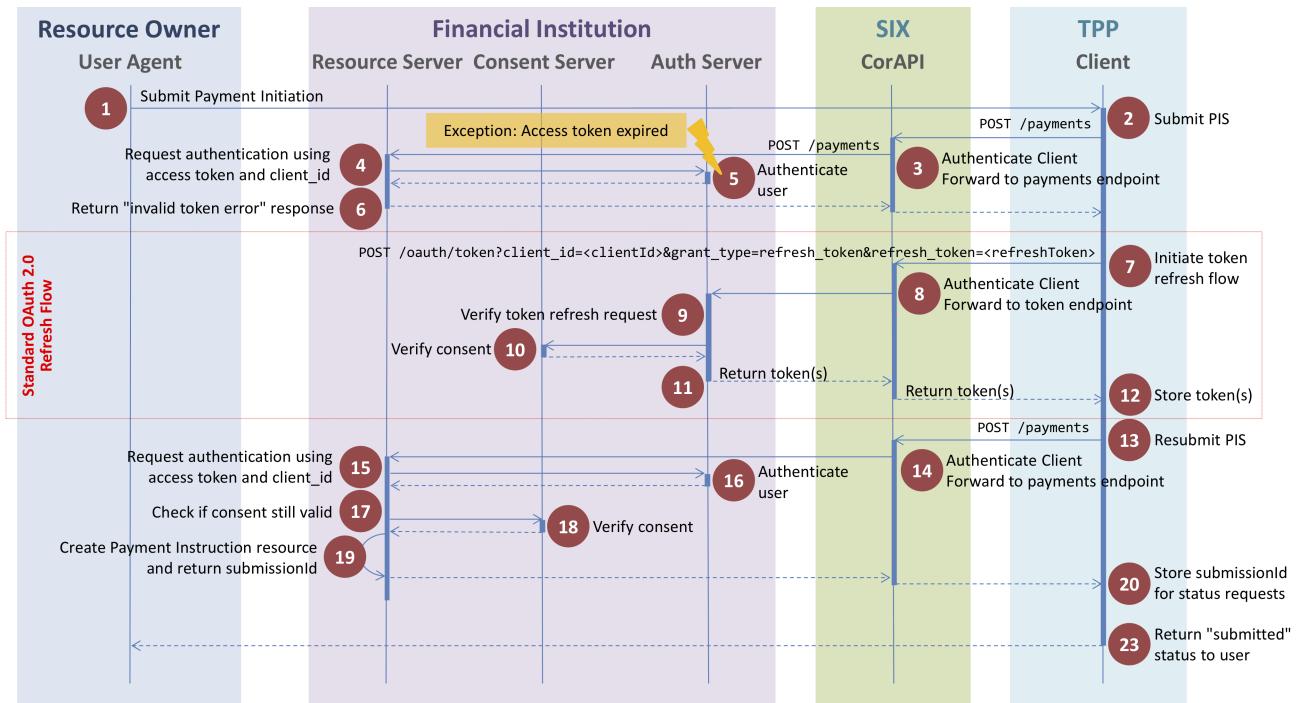


Figure 6. Overview of PIS flow with expired access token

1. Payment service user (PSU) is logged in at client and submits a payment initiation.
 2. TPP
 - a. stores IP address and User Agent of PSU
 - b. prepares PIS message
 - c. posts the message to Corporate API including **X-PSU-IP-Address** and **X-PSU-User-Agent** headers (mandatory for PIS)
- Example from client to SIX:

```

POST /api/bankingservices/corporate/v1/payments
X-CorAPI-Target-ID: 99999
Authorization: Bearer ZX1KMGVYQWlPaUpLVjFRaUxDShiR2Np
X-PSU-IP-Address: 34.12.19.3
X-PSU-User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/47.0.2526.111 Safari/537.36
...
Host: api-cert.six-group.com
  
```

3. Corporate API payments endpoint
 - a. authenticates the client
 - b. verifies authorization to payments endpoint
 - c. validates the request
 - d. forwards the request to the FI
- Example from SIX to FI:

```

POST /proprietaryBackendPath/v1/payments
X-CorAPI-Client-ID: 654321
Authorization: Bearer ZX1KMGVYQWlPaUpLVjFRaUxDSmhiR2Np
X-PSU-IP-Address: 34.12.19.3
X-PSU-User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/47.0.2526.111 Safari/537.36
...
Host: api.fi.example.com

```

4. Resource Server of FI extracts access token and client_id from PIS request and calls Auth Server to authenticate user.
5. Auth Server detects that access token is expired and returns an "invalid token error".
6. Resource Server takes "invalid token error" and returns it via Corporate API to the client.
7. TPP initiates standard OAuth 2.0 token refresh flow

Example from client to SIX:

```

POST /api/bankingservices/corporate/v1/oauth/token
X-CorAPI-Target-ID: 99999
client_id: 654321
grant_type: refresh_token
refresh_token: WWhRaU9qRTFOREV4TkRneE1Ea3NjbVY0
...
Host: api-cert.six-group.com

```

8. Corporate API token endpoint
 - a. authenticates the client
 - b. extracts the IID from the **X-CorAPITarget-ID** request header
 - c. determines the hostname/IP address and basepath of Auth Server token endpoint at target FI
 - d. sends token request with header **X-CorAPI-Client-ID** to target FI

Example from client to SIX:

```

POST /proprietaryBackendPath/v1/oauth/token
X-CorAPI-Client-ID: 654321
client_id: 654321
grant_type: refresh_token
refresh_token: WWhRaU9qRTFOREV4TkRneE1Ea3NjbVY0
...
Host: api.fi.example.com

```

9. Auth Server token endpoint of FI
 - a. authenticates TPP using request header **X-CorAPI-Client-ID**
 - b. verifies refresh_token

- c. calls Consent Server for consent verification
10. Consent Server verifies consent
11. Auth Server
- a. generates new access token
 - b. and either generates new refresh token or extends lifetime of refresh token
 - c. returns access token and optionally refresh token via Corporate API to the client.
If the request fails for any reason the Auth Server should inform the resource owner according to OAuth 2.0 specification, 5.2. [10].
12. TPP stores new access token (and optionally refresh token).
13. TPP resubmits the PIS message from step 2

Example from client to SIX:

```
POST /api/bankingservices/corporate/v1/payments
X-CorAPI-Target-ID: 99999
Authorization: Bearer 9pWlMxaVlXNXJhVzVuTFhWeIpYSXRhV1Fp
X-PSU-IP-Address: 34.12.19.3
X-PSU-User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/47.0.2526.111 Safari/537.36
...
Host: api-cert.six-group.com
```

14. Corporate API payments endpoint
- a. authenticates the client
 - b. verifies authorization to payments endpoint
 - c. validates the request
 - d. forwards the request to the FI

Example from SIX to FI:

```
POST /proprietaryBackendPath/v1/payments
X-CorAPI-Client-ID: 654321
Authorization: Bearer 9pWlMxaVlXNXJhVzVuTFhWeIpYSXRhV1Fp
X-PSU-IP-Address: 34.12.19.3
X-PSU-User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/47.0.2526.111 Safari/537.36
...
Host: api.fi.example.com
```

15. Resource Server of FI extracts access token and client_id from PIS request and calls Auth Server to authenticate user.
16. User is authenticated using access token and client_id.
17. Resource Server calls Consent Server to verify consent.

18. Consent Server verifies / returns consent.
19. Resource Server
 - a. stores/forwards **X-PSU-IP-Address** and **X-PSU-User-Agent** for fraud detection systems
 - b. creates PIS resource
 - c. returns response containing **submissionId** via Corporate API to TPP
20. TPP stores **submissionId** for consecutively polling the status of the PIS request.
21. TPP returns page showing "PIS submitted" status to user.

Chapter 8. Participants Directory

All participants (financial institutions and TPP's) using the Corporate API are listed in the directory. The directory holds the information necessary for operations and seamless integration of financial institutions and TPP's.

IMPORTANT The directory should not be replicated by the participants. However the data should be cached.

NOTE Quality assurance process by SIX has yet to be defined.

8.1. Access

Participants can access the directory through an API endpoint.

8.2. Management

The content of the directory is maintained by SIX.

8.3. Content

8.3.1. Participants

The information about the participants contains an unique identification (IID for financial institutions, clientId for TPPs), name, address, contact information (technical and business), logos/images and short/marketing descriptions.

All descriptions must be provided in German, English, French and Italian.

Images are supported in the following formats: PNG, JPEG and SVG. SVG is the preferred format for logos. The images are included base64-encoded.

IMPORTANT Participants are responsible to provide above information (except clientId provided by SIX).

Additionally the participant status, registration date and last modification date are stored.

Table 1. Participant status

Status	Description
ONBOARDING	The participant is in the onboarding process for Corporate API, potentially undergoing certification.
ACTIVE	The participant is certified and productive for at least one use case.
SUSPENDED	The participant has been temporarily suspended by SIX or by itself. Requests from/to suspended participants will be rejected by the platform.
INACTIVE	The participant is no longer active.

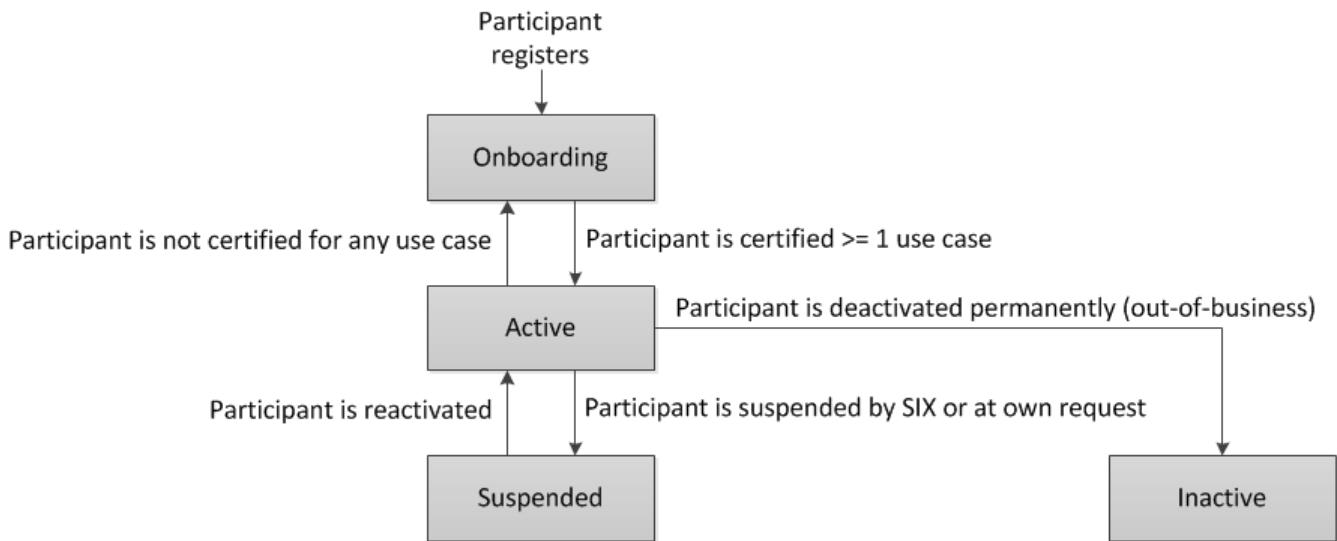


Figure 7. Status Diagram for Participant Directory

8.3.2. Software

The software used by a TPP is registered in the directory. This includes the product name, manufacturer, product category (i.e. accounting), product description, logos, etc.

The software used by financial institutions is not registered.

8.3.3. Use cases

This part of the directory describes the use cases supported by the participants. The use cases are versioned and their support is temporalized.

Use cases may contain additional properties to represent participant's specific implementation. E.g. `"maxXmlBulkItems: 100"` and `"maxXmlRequestSizeKb: 1024"` for use case XMLPMTS.

8.4. Deliverables by participants

All participants must provide the following information:

Table 2. Participant deliverables

Deliverable	Description	Mandatory
Company name	Name of legal entity	yes
Business contact	Email address and optionally phone	yes
Technical contact	Email address and optionally phone	yes
Company logo	SVG, JPG or PNG image of company logo	yes
Marketing image	SVG, JPG or PNG image (mood image)	no
Company URL	URL to participant's main website	yes

Deliverable	Description	Mandatory
Marketing description	Marketing text in form of max. 5 bullets describing the company's claim (not the product) in DE, EN, FR and IT	yes
Short description	Short description of the company in DE, EN, FR and IT	yes

Preferred image format for logos is SVG. If logos are in JPG or PNG format, two resolutions have to be provided: 100x100px and 500x500px

8.4.1. Additional deliverables for TPPs

TPPs must provide the following additional information:

Table 3. TPPs deliverables

Deliverable	Description	Mandatory
Consent backend URL	URL to be called by FIs to start the Preauthenticated OAuth Authorization Code flow (" <code>clientConsentBackendUrl</code> ")	yes
Redirection endpoint URL	Base URL for redirection in OAuth flow (" <code>clientRedirectionEndpointUrl</code> ",)	yes
Product name	Name of offered product/solution	yes
Categories	Categories of services the product offers. In MVP one sole category 'Accounting'	yes
Product info URL	URL to product description/specification	yes
Product logo	SVG, JPG or PNG image of product logo	yes
Marketing image	SVG, JPG or PNG image used to market product (mood image)	no
Marketing description	Marketing text in form of max. 5 bullets describing the product features (not the product) in DE, EN, FR and IT	yes
Short description	Short description of the product in DE, EN, FR and IT	yes

Preferred image format for logos is SVG. If logos are in JPG or PNG format, two resolutions have to be provided: 100x100px and 500x500px

8.4.2. Additional deliverables for FIs

FIs must provide the following additional information:

Table 4. FIs deliverables

Deliverable	Description	Mandatory	Scope
Auth Server URL	URL of the FI's auth server. Used by TPP to request an authorization code (" <code>providerAuthEndpointUrl</code> ")	yes	public
API hostname	Hostname of the FI's API (or reverse proxy)	yes	internal

Deliverable	Description	Mandatory	Scope
API port	Port number of the FI's API (or reverse proxy)	yes	internal
API IP address	IP address of the FI's API (or reverse proxy), required if hostname is not resolvable by public DNS	no	internal
API base path	Base path of the FI's API (or reverse proxy)	yes	internal
Token endpoint hostname	Hostname of the FI's token endpoint (or reverse proxy)	yes	internal
Token endpoint port	Port number of the FI's token endpoint (or reverse proxy)	yes	internal
Token endpoint IP address	IP address of the FI's token endpoint (or reverse proxy), required if hostname is not resolvable by public DNS	no	internal
Token endpoint base path	Base path of the FI's token endpoint (or reverse proxy)	yes	internal

Directory information with scope "internal" is not published to TPPs or FIs.

If FIs have multiple test environments, each environment requires a distinct entry in the directory.

Chapter 9. Glossary

Abbreviation	Definition	Explanation
TPP	Third Party Provider	#tbd
JSON	JavaScript Object Notation	#tbd
REST	Representational State Transfer	#tbd
RESTful API	#tbd	#tbd
API	Application Programming Interface	#tbd
SPS	Swiss Payment Standards	#tbd
SEPA	Single Euro Payment Area	#tbd
SME	Small and Medium-sized Enterprises	#tbd
AIS	Account Information Services	#tbd
PIS	Payment Initiation Services	#tbd
Swagger	#tbd	#tbd
OTP	One Time Password	#tbd
JWT	JSON Web Token	#tbd
IID	Institute Identification	#tbd
URI	Uniform Resource Identifier	#tbd
URL	Uniform Resource Locator	#tbd
XML	Extensible Markup Language	#tbd

Chapter 10. Appendix

10.1. Reason Codes

Code	Name	Definition	Note
AC02	InvalidDebtorAccountNumber	Debtor account number invalid or missing	Additional Info needed
AC03	InvalidCreditorAccount Number	Creditor account number invalid or missing	Additional Info needed
AM02	NotAllowedAmount	Specific transaction/message amount is greater than allowed maximum	-
AM03	NotAllowedCurrency	Specified message amount is an non processable currency outside of existing agreement	-
AM12	InvalidAmount	Amount is invalid or missing	-
BE04	MissingCreditorAddress	Specification of creditor's address, which is required for payment, is missing/not correct (formerly IncorrectCreditorAddress).	-
BE05	UnrecognisedInitiating Party	Party who initiated the message is not recognised by the end customer	-
BE11	InvalidCreditorCountry	Creditor country code is missing or invalid	-
BE22	MissingCreditorName	Creditor name is missing	-
CH16	ElementContentFormallyIncorrect	Content is incorrect	Additional Info needed
CH17	ElementNotAdmitted	Element is not allowed	Additional Info needed
CH20	DecimalPointsNotCompatibleWithCurrency	Number of decimal points not compatible with the currency	-
CH21	RequiredCompulsoryElementMissing	Mandatory element is missing	Additional Info needed

Code	Name	Definition	Note
CURR	IncorrectCurrency	Currency of the payment is incorrect	-
DUPL	DuplicatePayment	Payment is a duplicate of another payment	Additional Info needed
FF02	SyntaxError	Syntax error reason is provided as narrative information in the additional reason information.	Additional Info needed
FF10	BankSystemProcessing Error	File or transaction cannot be processed due to technical issues at the bank side	-
NARR	Narrative	Reason is provided as narrative information in the additional reason information.	Additional Info needed
RR07	RemittanceInformation Invalid	Remittance information structure does not comply with rules for payment type.	-
RR09	InvalidStructuredCredit orReference	Structured creditor reference invalid or missing.	-
RC04	InvalidCreditorBankIdentifier	Creditor bank identifier is invalid or missing	Additional Info needed
AG08	InvalidAccessRights	Transaction failed due to invalid or missing user or access right	-
AM21	LimitExceeded	Transaction amount exceeds limits agreed between bank and client.	-
CH03	RequestedExecutionDateOrRequestedCollectionDateTooFarInFuture	Value in Requested Execution Date or Requested Collection Date is too far in the future	-
DT01	InvalidDate	Invalid date (eg, wrong or missing settlement date)	-

Code	Name	Definition	Note
DT05	InvalidCutOffDate	Associated message, payment information block or transaction was received after agreed processing cut-off date, i.e., date in the past.	-
DU01	DuplicateMessageID	Message Identification is not unique.	-
DU02	DuplicatePaymentInformationID	Payment Information Block is not unique.	-

10.2. Allowed Certificates for platform access

The platform use various digital certificates to meet security and auditing requirements. SIX offers client certificates to access the platform.

In order to guarantee a high level of security, the certificates must meet the following conditions:

- **Validity:** Not expired
- **Version:** V3
- **Signature algorithm:** sha224RSA, sha256RSA, sha384RSA, sha512RSA
- **Key length:** At least RSA (2,048 bits)
- **Key usage:** Client authentication

The allowed list of issuers, can be found: [here](#)

Companies that want to use certificates from other third-party providers are asked to contact SIX.

10.3. Changelog

Title	What	Why
General	Corrections on typo and wording	formal correctness
General	Slightly changed endpoint descriptions	clarification
General	defined response scheme for ISO20022 endpoints	needed for code generation
General	Enum values changed from ESR to ISR	consistency
General	Removed dashes from technical Ids in examples	consistency
4.4 Payment Types	Payment Type 6 added	typo correction

Title	What	Why
4.5 Data Models	Update for all data models performed	feedback incorporation
4.6 Data Models	Update for all data models performed	feedback incorporation
4.7 Mapping to Schemes and Standards	Mapping updated	feedback incorporation
5.2 Changed ITAV to ITBD for balances information	feedback incorporation	5.2 Changed definitions to allow negative balance values
correction	5.3.4 Removed date from query parameters	correction
5.4 Added state diagram	clarification	5.5 Removed xmlelement from consent examples
correction	5.6.1 Changed parameter types to FormData	correction
6.1 Added requestedExecutionDate as mandatory field	correction	6.14 paymentCurrencyAmount
RegEx added	feedback incorporation	6.25 accountTransactionItemCharges
RegEx added	feedback incorporation	6.26 accountTransactionItemChargeSRecord
RegEx added	feedback incorporation	6.29 accountTransactionInstructedAmount
RegEx added	feedback incorporation	6.57 healthCheckResponse
Header names corrected and specified more details	feedback incorporation	8 Directory
Headers for Directory added	needed for code generation	8.3.3 Use Cases Directory
maxXmlBulkItems updated	feedback incorporation	9 Glossary
glossary updated	feedback incorporation	10 Added error code (RESOURCE_TO_LARGE)

10.4. Problem Definitions

10.4.1. Bad Request (400)

The payload was not valid

/problems/INVALID_PAYLOAD

ID is missing

The payload is malformed

/problems/MALFORMED_PAYLOAD

Malformed JSON

10.4.2. Unauthorized (401)

The oAuth Token is invalid

/problems/INVALID_TOKEN

Something is wrong with this token

The oAuth Token is expired

/problems/EXPIRED_TOKEN

The token is no longer valid

10.4.3. Forbidden (403)

No privileges for the requested operation

/problems/INSUFFICIENT_PRIVILEGES

Insufficient privileges for the requested operation

The provided token does not grant access to the requested account

/problems/NO_ACCESS_TO_RESOURCE

More details

10.4.4. Not Found (404)

The requested resource does not exist

/problems/RESOURCE_DOES_NOT_EXIST

More details

The requested resource is not yet available

/problems/RESOURCE_NOT_READY

Resource xyz will be available in 00:05:00

10.4.5. Not Allowed (405)

This HTTP Operation is not allowed on this endpoint

/problems/WRONC_METHOD

Only GET operations are allowed

This operation is not allowed for the specified resource

/problems/OPERATION_NOT_ALLOWED

Reason

10.4.6. Technical Errors (500)

Technical error on server side

/problems/TECHNICAL_ERROR

Processing yielded a technical error.

Generated resource was to large

/problems/RESOURCE_TOO_LARGE

The generated resource exceeded the size limit

10.5. Open Points

There are some open topics that are addressed but not yet finalized:

- API and use case versioning
- Glossary