OPEN BANKING CONNECT

API Specification Release 1.18.0

16 JULY 2020

mastercard

HIGHLY CONFIDENTIAL

Authorisation

To the extent that is not already in the public domain, this document and its contents, which includes all materials, concepts and ideas, is strictly confidential, private and personal to Mastercard and shall not be used, copied, distributed or reproduced in whole or in part, in any form, to any third party without prior consent of Mastercard. The copyright in this document is owned by Mastercard. The purpose of this document is to provide a customer with an overview of the Mastercard Open Banking (OB) Connect API Specification changes for the stated release. This document is intended for reading primarily by OB development and integration staff and by administration personnel. They will use it as a guide for creating and maintaining the OB API message interchange. The information outlined in this document shall only be used for the purpose for which it was originally provided.

Please refer any questions regarding the above to:

Michael Clark

Open Banking Product Manager

Mastercard

1 Angel Lane

EC4R 3AB

London

Michael.clark2@mastercard.com

© Mastercard 2020. All rights reserved.

Contents

1 About this guide 7
1.1 Document Purpose 7
1.2 Scope 7
1.3 Glossary 7
2 Open Banking Connect Overview 8
2.1 Open Banking Background 8
2.2 MAOB Connect Hub 8
2.3 Special considerations 8
2.4 Accessing the Open Banking Connect APIs 9
2.5 Important Notice for Current Single API Users 9
3 Notations 10
3.1 Conditions 10
3.2 Multiplicity 10
3.3 Interface API structure 10
3.4 API Profiles 11
4 Business context diagrams 12
4.1 AIS – Submit Consent 12
4.2 AIS – Exchange the PSU authorization for access consent 13
4.3 AIS – Get List of Accounts 14
4.4 AIS – Get Account Details 15
4.5 AIS – Get Account Balance 16
4.6 AIS – Get Account Transactions 17
4.7 AIS – Get Account Transaction Details 18
4.8 AIS – Get Raw Consent 19
4.9 AIS – Delete Account Information Consent 20
4.10 PIS - Domestic Credit Transfer – Submit Consent 21
4.11 PIS - Domestic Credit Transfer – Submit Payment Initiation 22
4.12 PIS - Domestic Credit Transfer – Get Payment Status 23
4.13 PIS - SEPA Credit Transfer – Submit Consent 23
4.14 PIS - SEPA Credit Transfer – Submit Payment Initiation 24
4.15 PIS - SEPA Credit Transfer – Get Payment Status 26
4.16 PIS – Cross-Border Credit Transfer – Submit Consent 26
4.17 PIS - Cross-Border Credit Transfer – Submit Payment Initiation 28
4.18 PIS - Cross-Border Credit Transfer – Get Payment Status 29
5 API Access Methods 30
5.1 Accounts endpoints 30
5.2 Payments endpoints 31
6 Response codes and Error Handling 32
6.1 HTTP response codes 32

6.2 Error information 32
6.3 Error codes 33
6.3.1 HTTP Status 500/504 - General reason codes 33
6.3.2 HTTP Status 400 - General reason codes 34
6.3.3 HTTP Status 400 - AIS - Account Information Consent requests - Specific reason codes 37
6.3.4 HTTP Status 400 - AIS – Get List of Accounts – Specific reason codes 38
6.3.5 HTTP Status 400 - AIS – Exchange the PSU authorization for access consent requests - Specific reason codes 39
6.3.6 HTTP Status 400 - AIS – Get Account Details – Specific reason codes 40
6.3.7 HTTP Status 400 - AIS – Get Account Balances – Specific reason codes 41
6.3.8 HTTP Status 400 - AIS - Get Account Transactions - Specific reason codes 42
6.3.9 HTTP Status 400 - AIS – Get Transaction Details – Specific reason codes 46
6.3.10 HTTP Status 400 - PIS - Domestic Payment Initiation Consent request - Specific reason codes 47
6.3.11 HTTP Status 400 - PIS - Domestic Payment Initiation Transfer request - Specific reason codes 57
6.3.12 HTTP Status 400 - PIS - Domestic Payment Status request - Specific reason codes . 58
6.3.13 HTTP Status 400 – PIS - SEPA Payment Initiation Consent request - Specific reason codes 58
6.3.14 HTTP Status 400 - PIS- SEPA Payment Initiation Transfer request - Specific reason codes 64
6.3.15 HTTP Status 400 - PIS- SEPA Payment Status request - Specific reason codes 64
6.3.16 HTTP Status 400 - PIS- Cross-border Payment Initiation Consent request - Specific reason codes 64
6.3.17 HTTP Status 400 - PIS– Cross-border Payment Initiation Transfer request - Specific reason codes 68
6.3.18 HTTP Status 400 - Payment Initiation Service – Cross-border Payment Status request - Specific reason codes 68
7 Account Information Service 69
7.1 Get List of ASPSPs Request 69
7.2 Get Status of Inbound API and AIS Connectors 74
7.3 Get Raw Consent Request 76
7.4 Account Information Consent Request 80
7.5 Exchange the PSU Authorization for Access Consent Request 85
7.6 Get List of Accounts Request 90
7.7 Get Account Details Request 96
7.8 Get Account Balances Request 103
7.9 Get Account Transactions Request 109
7.10 Get Transaction Details Request 121
7.11 Delete Account Information Consent Request 132
8 Payment Initiation Service 136
8.1 Get List of ASPSPs Request 136
8.2 Get Status of Inbound API and PIS Connectors 140
8.3 Get Raw Consent Request 142
8.4 Domestic Payments 146
8.4.1 Payment Initiation Consent Request 146

8.4.2 Payment Credit Transfer Initiation Request 160
8.4.3 Get Payment Status Request 166
8.5 SEPA Payments 171
8.5.1 Payment Initiation Consent Request 171
8.5.2 Payment Credit Transfer Initiation Request 184
8.5.3 Get Payment Status Request 189
8.6 Cross-Border (International) Payments 194
8.6.1 Payment Initiation Consent Request 194
8.6.2 Payment Credit Transfer Initiation Request 208
8.6.3 Get Payment Status Request 214
9 Appendix 219

Revision History

Revision	Date of revision	Author	Description of modifications
0.1	15 July 2020	Mastercard Development / Mastercard CS&D / Michael Clark	First draft for Release
1.0	16 July 2020	Mastercard CS&D	Published document

1 About this guide

1.1 Document Purpose

This guide provides the Mastercard Open Banking (MAOB) Connect Hub customers with the information required to use the MAOB Connect Hub universal API (Application Program Interface). In conjunction with the respective YAML (YAML Ain't Markup Language) specification, this guide provides a full reference of the API to ensure a successful integration with the hub.

This document is intended for primarily Open Banking (OB) development, integration staff, and administration personnel. They will use this as a guide for creating and maintaining OB API message interchange.

1.2 Scope

This version covers the initial (Pilot) version of the API (which is currently in development) and is subject to review/update as and when the respective API implementation will evolve. Mastercard released this version (Release1.18) and it includes all previous releases not implemented for a specific client.

1.3 Glossary

Table 1: Acronyms and terms

Acronym or term	Description
AISP	Account Information Service Provider
API	Application Program Interface
ASPSP	Account Servicing Payment Services Provider – a financial institution/PSP servicing the Accounts of the PSU
Consent	The process by which a PSU gives a TPP permission to approach their ASPSP in order that the TPP can gain access to the PSU's account to provide the service to the PSU
Consent Authorisation	The process by which the PSU confirms that the ASPSP may respond to the request from the TPP to whom the PSU has given consent
MA	Mastercard
MAOB Connect	Mastercard Open Banking Connectivity Hub – the service that translates TPP requests and routes them through one of the supported ASPSP APIs
PISP	Payment Initiation Service Provider
PSU	Payment Services User - the holder of a current personal or business account at the ASPSP. PSU wants to use services provided by the TPP.
SCA	Strong Customer Authentication
TPP	Third Party Provider - the regulated entity that requires access to the PSU account information from ASPSP in order to provide them services in accordance with PSD2
URL	Uniform Resource Locator - the address of Internet resource
YAML	YAML Ain't Markup Language is a human-readable data serialization language.

2 Open Banking Connect Overview

The Mastercard Open Banking Connect product provides access to Financial Institutions' (FI) user account information and payment initiation API capabilities in a reliable and secure manner to Third Party Provider (TPP) applications.

2.1 Open Banking Background

The Payment Service Directive (PSD) known as PSD2 (PSD2, Directive (EU) 2015/2366), issued by the European Union (EU), allows regulated TPPs, with the consent of the Payment Services User (PSU), to connect to customer accounts held by Account Servicing Payments Services Providers (ASPSPs). This connection is in order to securely obtain account information or initiate payments to or from the respective accounts.

The ASPSPs give access to three types of services to implement the APIs used to allow the connection:

- Payment Initiation Service (PIS) – Operated by a Payment Initiation Service Provider (PISP) TPP, as defined by article 66 of PSD2.
- Account Information Service (AIS) – Operated by an Account Information Service Provider (AISP) TPP, as defined by article 67 of PSD2.
- Confirmation of the Availability of Funds service (CAF) – Used by Payment Instrument Issuing Service Provider (PIISP) TPP, as defined by article 65 of PSD2.

PSD2 does not specify the actual API standards used by ASPSPs. ASPSPs can select any of the standards available on the market already or implement their own, as long as it complies with the provisions of PSD2. Thus, integrating a TPP with even a small number of ASPSPs may require a significant amount of effort and investment.

2.2 MAOB Connect Hub

The Connect Hub provides a seamless integration path with ASPSPs, even if using different APIs. It provides a single point of entry API that acts as a gateway that translates the messages received from TPPs and routes them to and from a list of supported ASPSPs.

See the full list of operations in section 5 API Access Methods. Additional details are also available in sections 7 Account Information Service and 8 Payment Initiation Service.

2.3 Special considerations

To create the APIs for this product, Mastercard leveraged the Berlin Group specification standard as an industry reference (alongside other global standards) for Mastercard's own data model, per the Berlin Group Creative Commons Attribution 4.0 International Public License:

https://www.berlin-group.org/nextgenpsd2-downloads

2.4 Accessing the Open Banking Connect APIs

Users can access the Mastercard Open Banking Connect Hub via Mastercard Developer at https://developer.mastercard.com/product/open-banking-connect. Through the developer portal, users can access documentation, interact with sandboxed versions of the APIs, manage credentials, and begin the process of setting up a production application.

2.5 Important Notice for Current Single API Users

Prior to the Drop 2 (June 2019) release of the Open Banking Connect API Specification, there was only a single API. Due to a need to split the AIS and PIS functions in Drop 2, a second API has been introduced for AIS. PIS will continue to use the original API.

Customers that want to use both AIS and PIS functions must connect to both APIs.

Current customers on a single API must contact their Onboarding contact or their Open Banking Development team contact to ensure they are correctly pointing to both the APIs.

New customers must use the Mastercard Developer site to access the APIs.

3 Notations

3.1 Conditions

Each message specification data field/element includes the following condition codes. Each condition is described in the respective request or response.

- O (Optional): The field is supported by the API, but the usage is optional.
- C (Conditional): The field is required by the API under certain conditions (see "Description").
- M (Mandatory): The field is always required by the API.

3.2 Multiplicity

Multiplicity indicates the cardinality of the elements in the message structure:

- [0..1] – 0 or 1 instances of the element
- [1..1] – exactly one instance of the element only
- [0..N] – 0 to N instances of the element

3.3 Interface API structure

The API Interface is resource oriented. Resources are addressed under the API endpoints. For example:

```
https://{provider}/{service}
```

Where

- `{provider}` is the host and path of the API
- `{service}` has the values consents, payments, accounts, eventually extended by more information on product types and request scope

3.4 API Profiles

All ASPSPs that can be accessed through Mastercard Open Banking Connect have a descriptor that indicates the API Profile of the ASPSP, that is, the Open Banking API standard implemented on the side of the ASPSP. As of the Drop 2 release, the MAOB Connect provides access only to ASPSPs that have one of the following API profiles:

- **CMA9** – An Open Banking API standard defined by the Open Banking Implementation Entity (OBIE) in UK, which is mandatory for the nine major banks, per the order from Competition and Markets Authority (CMA). Currently this API profile is based on v2.0.0 of the OBIE specification for AIS and on v1.1.0 of the OBIE specification PIS.
- **PolishAPI** – An Open Banking API standard for the banking sector in Poland currently based on v2.1.2 of the Polish API specification.
- **NextGenPSD2** – An Open Banking API standard defined by Berlin Group for the EU banking sector. They developed the standard to create uniform and interoperable communications between EU banks and TPPs.
- **STET** - An Open Banking API standard defined by group of major French banks for full range of payment instruments in the French and Belgian market

The API Profile enables the TPP to determine if the original API standard of the ASPSP supports a specific endpoint included in the MAOB Connect. The endpoints specifications included in this document provide details of the supported API Profile.

The API returns the API Profile for each ASPSP when the TPP requests the list of ASPSPs.

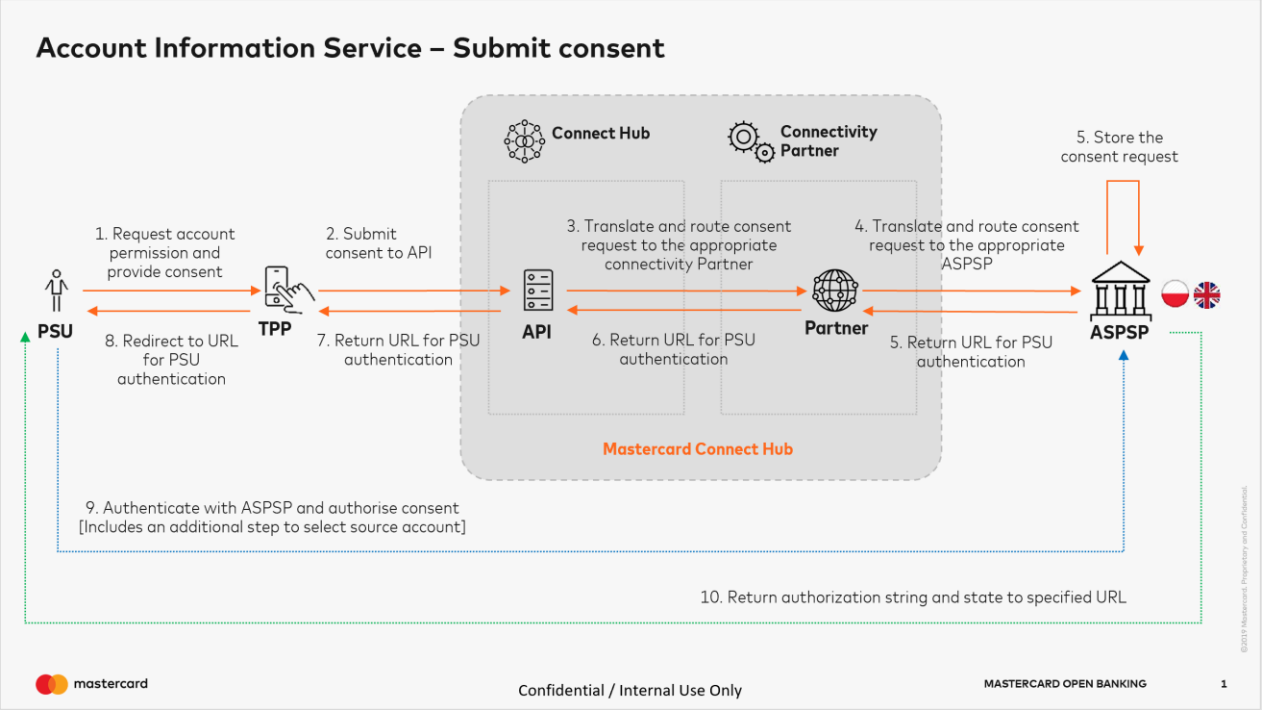
4 Business context diagrams

The Business Context Diagrams provide an overview of the interactions between external business role-players and MAOB Connect HUB. The following example Business Context Diagrams use the Account Information Service and Domestic Transfer endpoints. All other service endpoints follow similar message flows.

4.1 AIS – Submit Consent

The following Business Context Diagram shows the flows used to initiate the creation of consent for the Account Information operations.

Image 1: Account Information Service – Submit consent



To initiate an Account Information operation, the following actions are performed:

1. The Payment Services User (PSU) populates account details and provides consent to the TPP.
2. The TPP submits consent to MAOB Connect Hub using MAOB API.
3. MAOB API translates and routes a consent request to the appropriate Connectivity Partner.
4. The Connectivity Partner translates and routes the consent request to the appropriate ASPSP.
5. The ASPSP stores the consent request and returns the URL for the PSU authentication to the Connectivity Partner.
6. The Connectivity Partner returns URL for PSU authentication to MAOB API.
7. The MAOB API returns URL for PSU authentication to the TPP.
8. The TPP redirects the PSU to the URL for PSU authentication.
9. The PSU authenticates with the ASPSP and authorizes consent.
10. The ASPSP returns the authorization string and state to the specified URL.

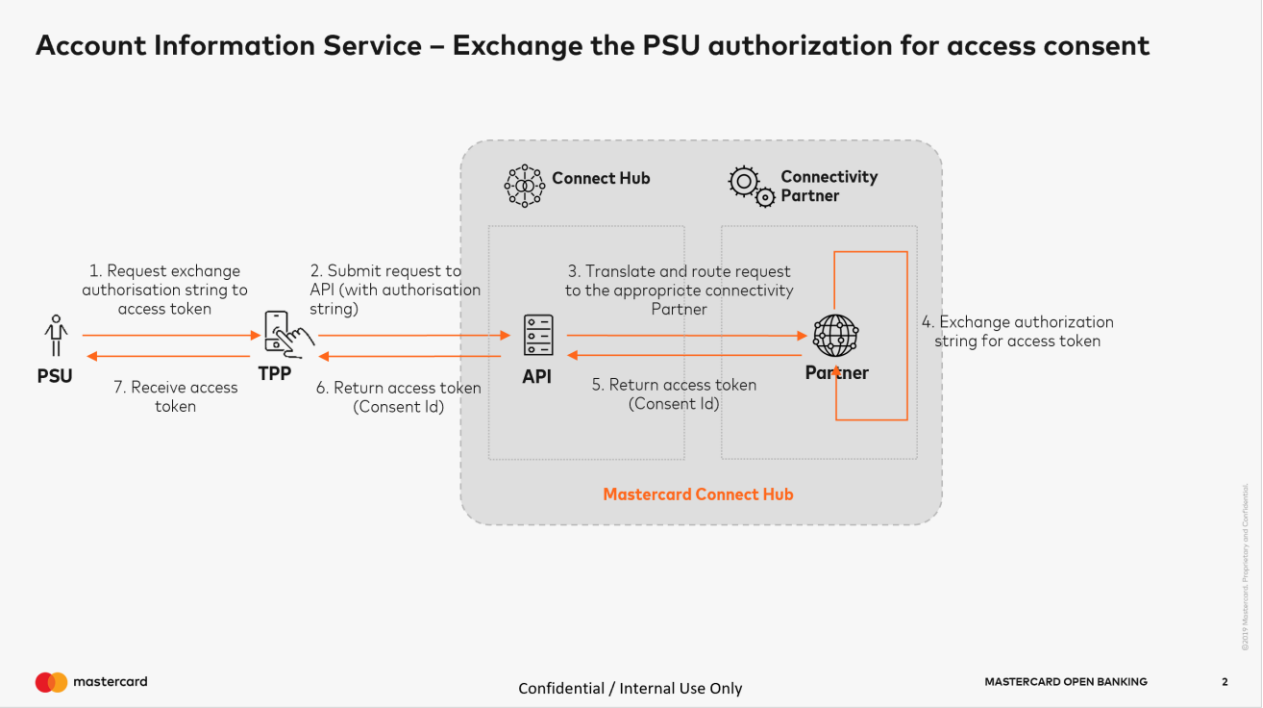
4.2 *AIS – Exchange the PSU authorization for access consent*

The following Business Context Diagram shows the flows used to exchange the PSU authorization string for access token on the Connectivity Partner side.

Prerequisite: PSU should perform the following operations before the 'Exchange the PSU authorization for access consent' call is executed:

1. 'Submit Consent' operation to get authorization string

Image 2: Account Information Service – Exchange the PSU authorization string for access token



The following actions are performed to initiate an exchange authorization string for access token:

1. The PSU send authorization string provided by the ASPSP for access token exchange.
2. The TPP submits the exchange request with authorization string to MAOB Connect Hub using MAOB API.
3. MAOB API translates and routes the request to the appropriate Connectivity Partner.
4. The Connectivity Partner exchanges authorization string for access token (Consent Id).
5. Connectivity Partner returns access token (Consent Id) to MAOB API.
6. MAOB API returns access token (Consent Id) to the TPP.
7. PSU receives access token (Consent Id).

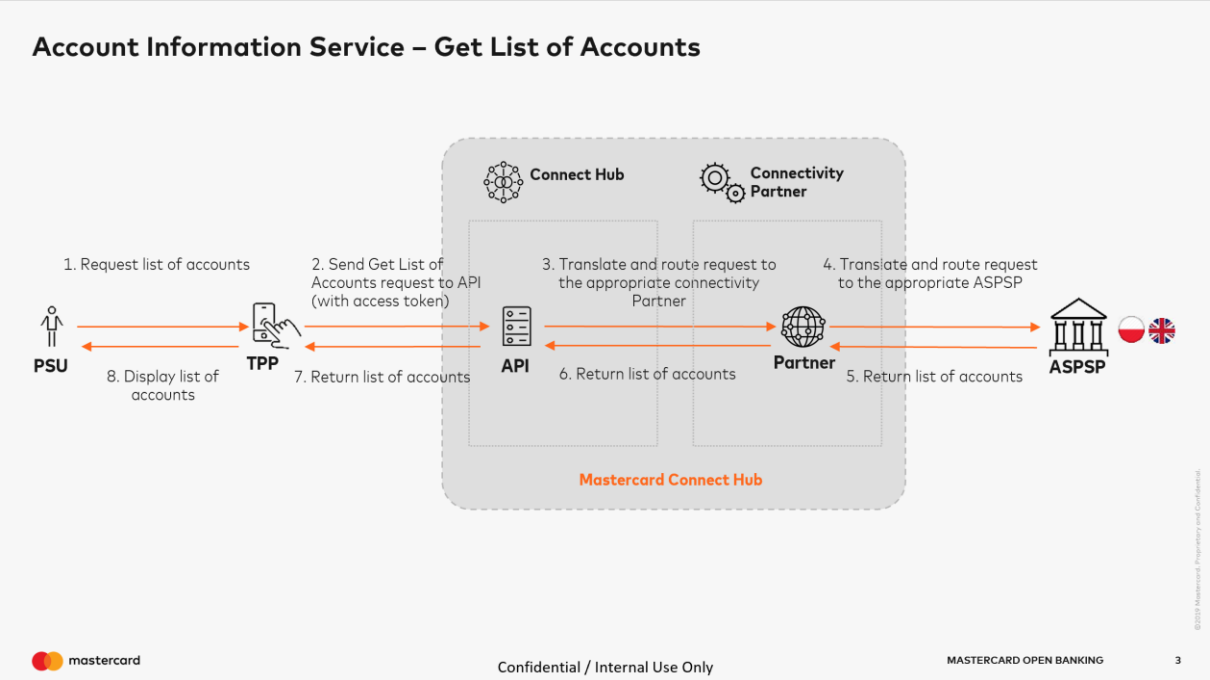
4.3 *AIS – Get List of Accounts*

The following Business Context Diagram shows the flows used to retrieve the list of accounts from the ASPSP based on previously obtained access token.

Prerequisites: PSU should perform the following operations before the 'Get List of Accounts' call is executed:

1. 'Submit Consent' operation to get authorization string
2. 'Exchange PSU authorization for access consent' operation to get access token

Image 3: Account Information Service – Get List of Accounts



To perform a Get List of Accounts request, the following actions are performed:

1. The PSU sends a get list of accounts request to the TPP.
2. The TPP submits the request with previously obtained access token (Consent Id) to MAOB Connect Hub using MAOB API.
3. MAOB API translates and routes the request to the appropriate Connectivity Partner.
4. The Connectivity Partner translates and routes the request to the appropriate ASPSP.
5. The ASPSP returns the list of accounts to Connectivity Partner.
6. The Connectivity Partner returns the list of accounts to MAOB API.
7. MAOB API returns the list of accounts to the TPP.
8. The TPP displays the list of accounts to PSU.

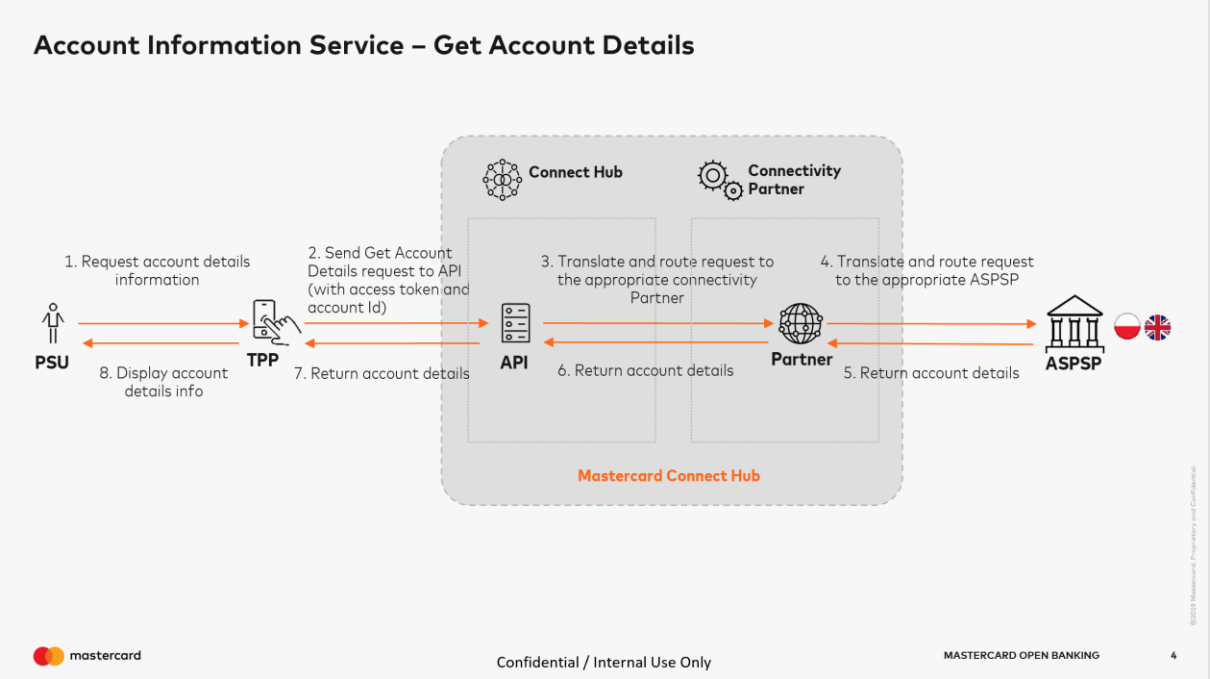
4.4 *AIS – Get Account Details*

The following Business Context Diagram shows the flows used to retrieve the account details information from the ASPSP based on previously obtained access token and account identification code.

Pre-requests: PSU should perform the following operations before the 'Get Account Details' call is executed:

1. 'Submit Consent' operation to get authorization string
2. 'Exchange PSU authorization for access consent' operations to get access token
3. 'Get List of Accounts' operation to get account identifier

Image 4: Account Information Service – Get Account Details



To perform a Get Account Details request, the following actions are performed:

1. The PSU sends a get accounts details request to the TPP.
2. The TPP submits the request with previously obtained access token (Consent Id) and account identifier (Account Id) to MAOB Connect Hub using MAOB API.
3. MAOB API translates and routes the request to the appropriate Connectivity Partner.
4. The Connectivity Partner translates and routes the request to the appropriate ASPSP.
5. The ASPSP returns the account details to Connectivity Partner.
6. The Connectivity Partner returns the account details to MAOB API.
7. MAOB API returns the account details to the TPP.
8. The TPP displays the account details information to PSU.

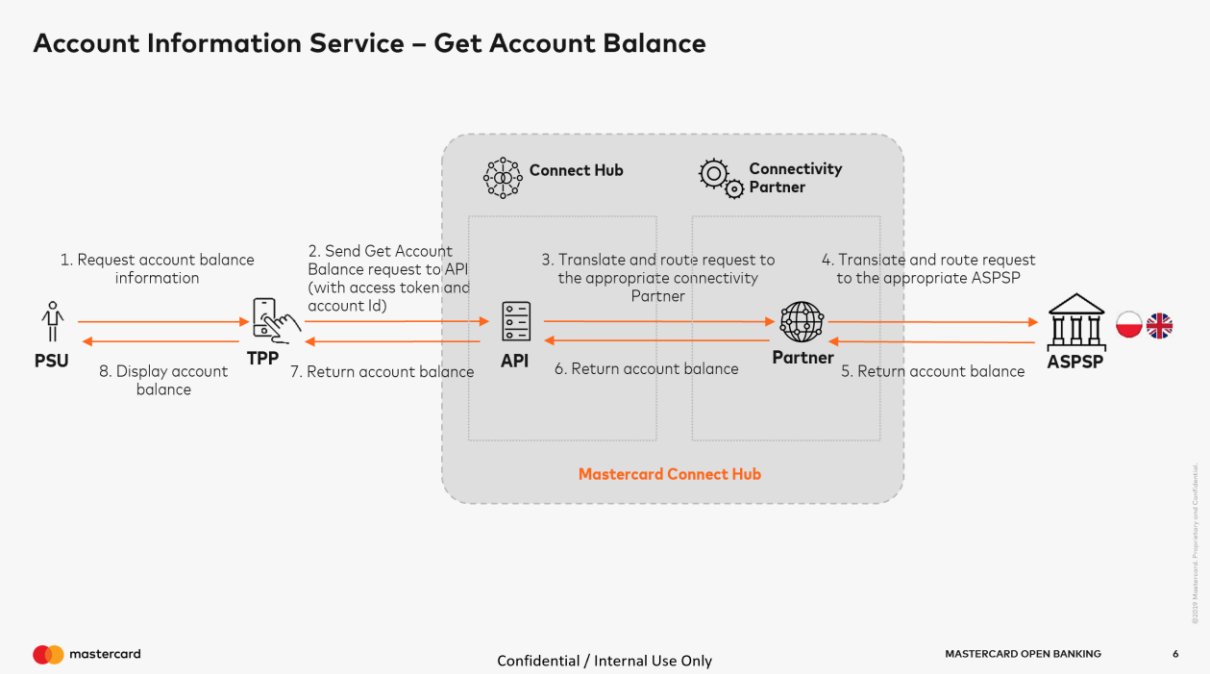
4.5 *AIS – Get Account Balance*

The following Business Context Diagram shows the flows used to retrieve the account balance information from the ASPSP based on previously obtained access token and account identification code.

Pre-requests: PSU should perform the following operations before the 'Get Account Balance' call is executed:

1. 'Submit Consent' operation to get authorization string
2. 'Exchange PSU authorization for access consent' operations to get access token
3. 'Get List of Accounts' operation to get account identifier

Image 5: Account Information Service – Get Account Balance



To perform a Get Account Balance request, the following actions are performed:

1. The PSU sends a get accounts balance request to the TPP.
2. The TPP submits the request with previously obtained access token (Consent Id) and account identifier (Account Id) to MAOB Connect Hub using MAOB API.
3. MAOB API translates and routes the request to the appropriate Connectivity Partner.
4. The Connectivity Partner translates and routes the request to the appropriate ASPSP.
5. The ASPSP returns the account balance information to Connectivity Partner.
6. The Connectivity Partner returns the account balance to MAOB API.
7. MAOB API returns the account balance to the TPP.
8. The TPP displays the account balance information to PSU.

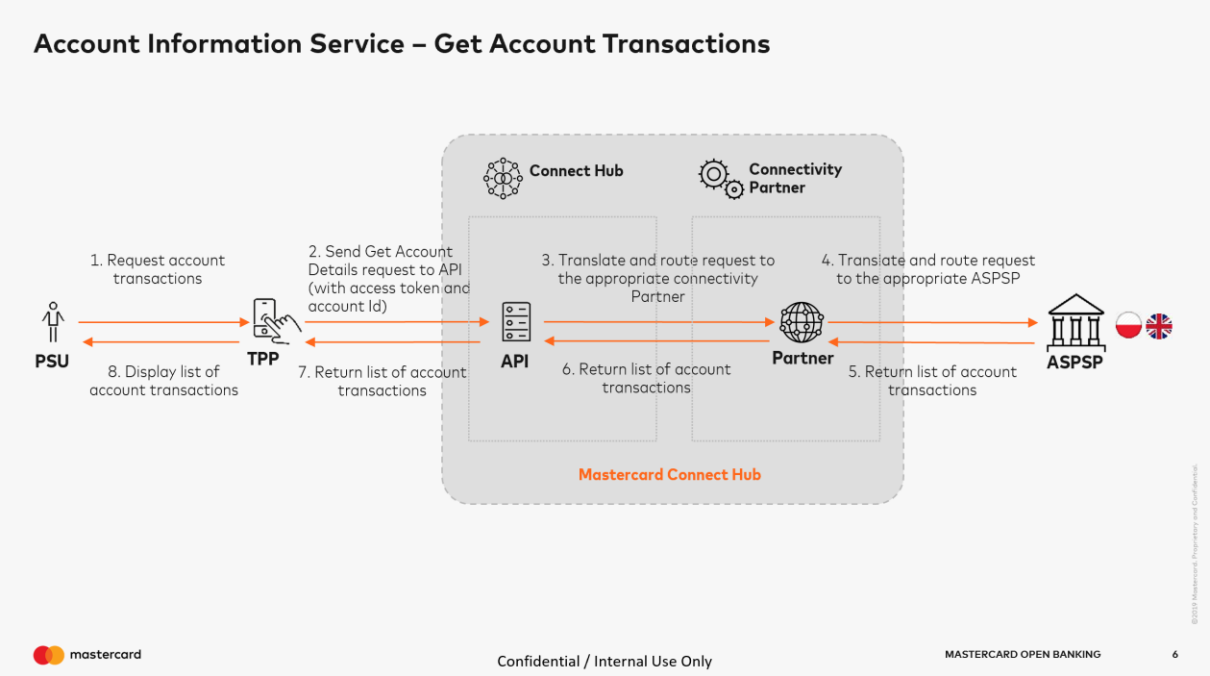
4.6 *AIS – Get Account Transactions*

The following Business Context Diagram shows the flows used to retrieve the list of account transactions from the ASPSP based on previously obtained access token and account identification code.

Pre-requests: PSU should perform the following operations before the 'Get Account Transactions' call is executed:

1. 'Submit Consent' operation to get authorization string
2. 'Exchange PSU authorization for access consent' operations to get access token
3. 'Get List of Accounts' operation to get account identifier

Image 6: Account Information Service – Get Account Transactions



To perform a Get Account Transactions request, the following actions are performed:

1. The PSU sends a get accounts transactions request to the TPP.
2. The TPP submits the request with previously obtained access token (Consent Id) and account identifier (Account Id) to MAOB Connect Hub using MAOB API.
3. MAOB API translates and routes the request to the appropriate Connectivity Partner.
4. The Connectivity Partner translates and routes the request to the appropriate ASPSP.
5. The ASPSP returns the list of account transactions to Connectivity Partner.
6. The Connectivity Partner returns the list of account transactions to MAOB API.
7. MAOB API returns the list of account transactions to the TPP.
8. The TPP displays the account transactions list to PSU.

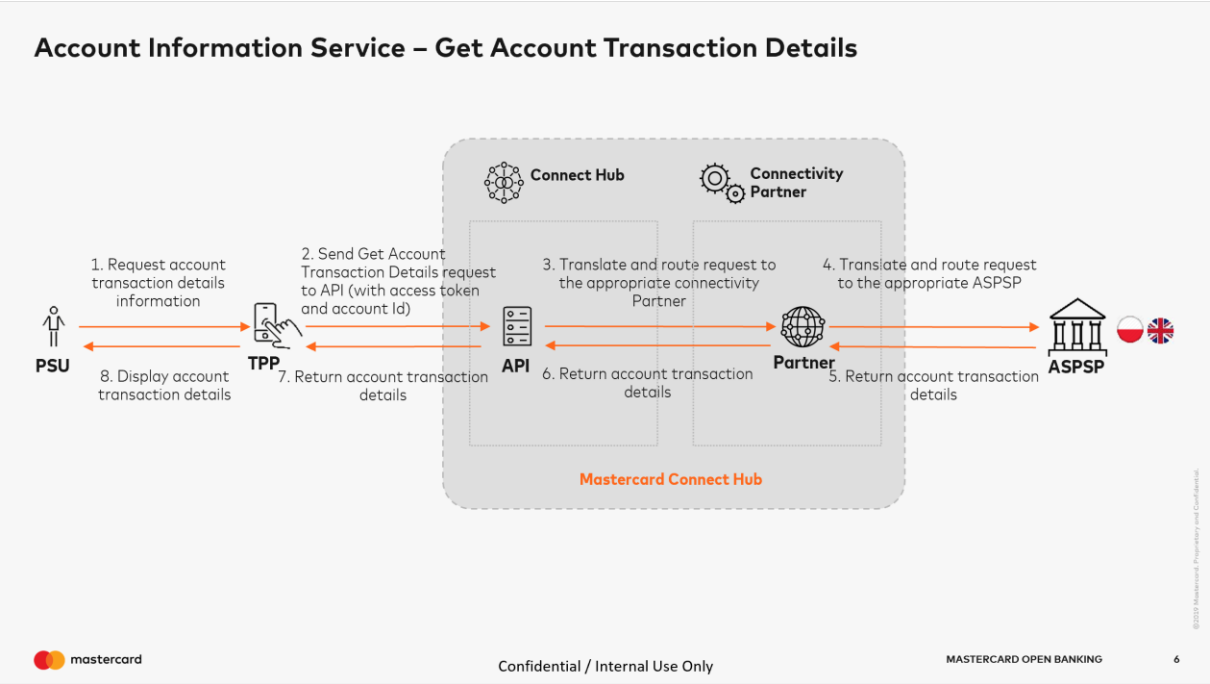
4.7 AIS – Get Account Transaction Details

The following Business Context Diagram shows the flows used to retrieve account transaction details from the ASPSP based on previously obtained access token and transaction identification code.

Pre-requests: PSU should perform the following operations before the 'Get Account Transaction Details' call is executed:

1. 'Submit Consent' operation to get authorization string
2. 'Exchange PSU authorization for access consent' operations to get access token
3. 'Get List of Accounts' operation to get account identifier
4. 'Get Account Transactions' operation to get transaction identifier

Image 7: Account Information Service – Get Account Transaction Details



To perform a Get Account Transaction Details request, the following actions are performed:

1. The PSU sends a get account transaction details request to the TPP.
2. The TPP submits the request with previously obtained access token (Consent Id) and transaction identifier (Transaction Id) to MAOB Connect Hub using MAOB API.
3. MAOB API translates and routes the request to the appropriate Connectivity Partner.
4. The Connectivity Partner translates and routes the request to the appropriate ASPSP.
5. The ASPSP returns the account transaction details to Connectivity Partner.
6. The Connectivity Partner returns the account transaction details to MAOB API.
7. MAOB API returns the account transaction details to the TPP.
8. The TPP displays the account transaction details to PSU.

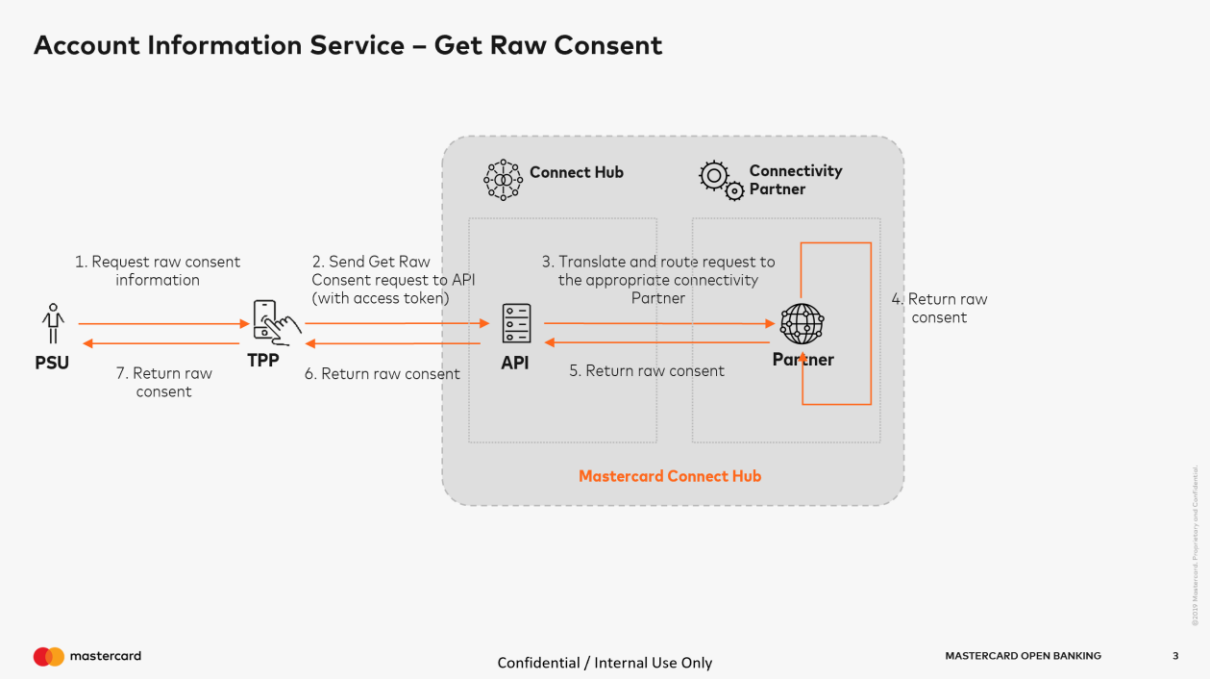
4.8 AIS – Get Raw Consent

The following Business Context Diagram shows the flows used to get raw consent from the Connectivity Partner side.

Pre-requests: PSU should perform the following operations before the 'Get Raw Consent' call is executed:

1. 'Submit Consent' operation to get authorization string
2. 'Exchange PSU authorization for access consent' operation to get access token

Image 8: Account Information Service – Get Raw Consent



The following actions are performed to initiate a Get Raw Consent request:

1. The PSU send a get raw consent request with previously obtained access token (Consent Id) to the TPP.
2. The TPP submits the get raw consent request to MAOB Connect Hub using MAOB API.
3. MAOB API translates and routes the request to the appropriate Connectivity Partner.
4. The Connectivity Partner returns raw consent
5. The Connectivity Partner returns raw consent to MAOB API
6. MAOB API returns raw consent to the TPP.
7. TPP returns raw consent to the PSU

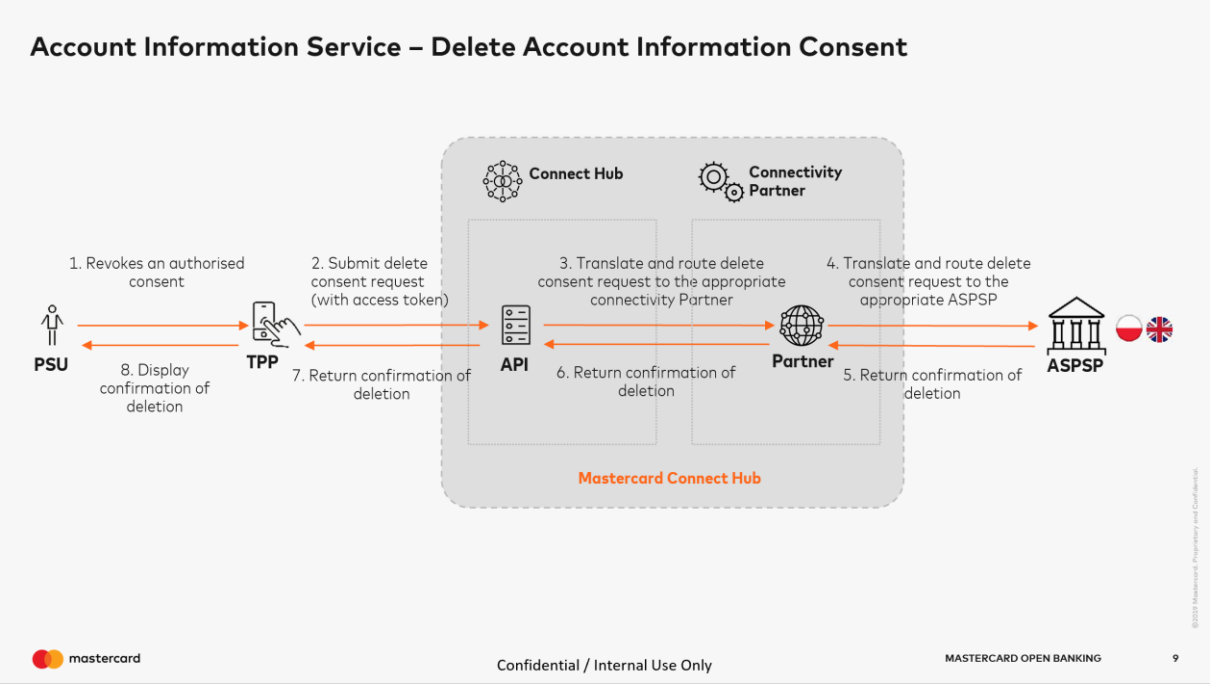
4.9 *AIS – Delete Account Information Consent*

The following Business Context Diagram shows the flows used to delete account information consent from the ASPSP based on previously obtained access token.

Pre-requests: PSU should perform the following operations before the 'Delete Account Information Consent' call is executed:

1. 'Submit Consent' operation to get authorization string
2. 'Exchange PSU authorization for access consent' operation to get access token

Image 9: Account Information Service – Delete Account Information Consent



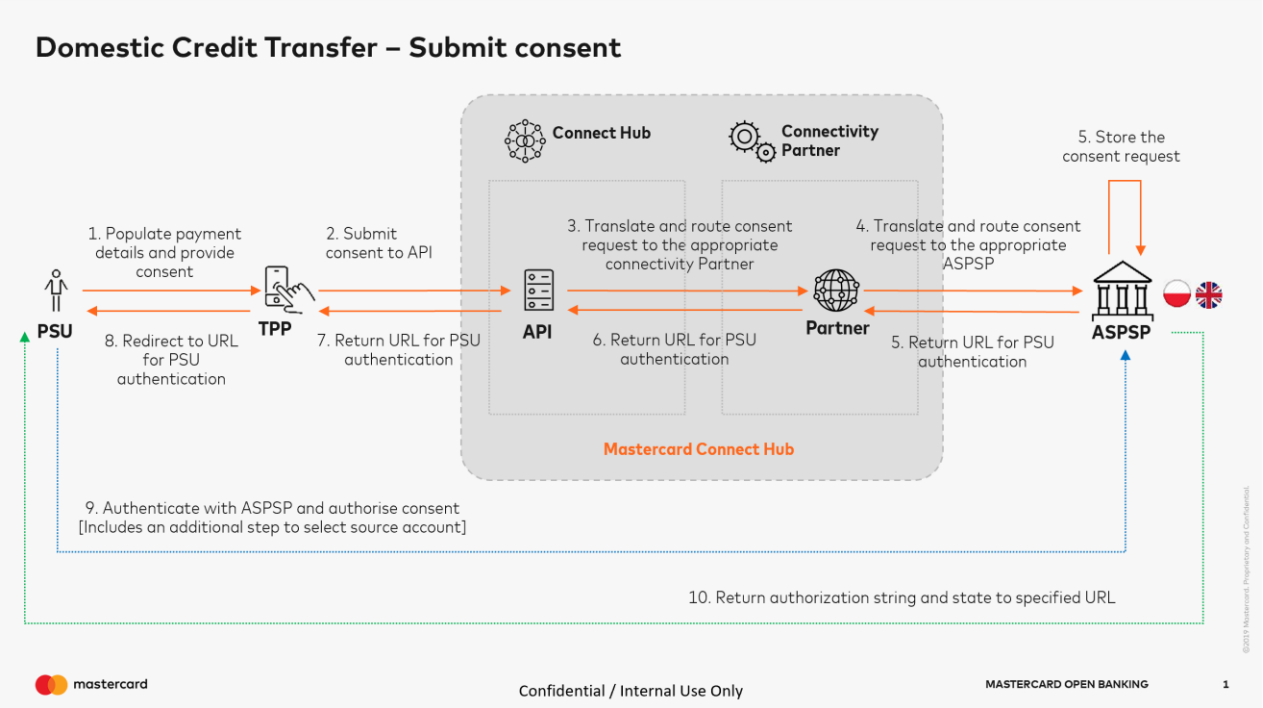
To perform a Delete Account Information Consent request, the following actions are performed:

1. The PSU sends a delete account information consent request to the TPP.
2. The TPP submits the request with previously obtained access token (Consent Id) to MAOB Connect Hub using MAOB API.
3. MAOB API translates and routes the request to the appropriate Connectivity Partner.
4. The Connectivity Partner translates and routes the request to the appropriate ASPSP.
5. The ASPSP returns confirmation of deletion to Connectivity Partner.
6. The Connectivity Partner returns confirmation of deletion to MAOB API.
7. MAOB API returns confirmation of deletion to the TPP.
8. The TPP displays confirmation of deletion to PSU.

4.10 PIS - Domestic Credit Transfer – Submit Consent

The following Business Context Diagram shows the flows used to initiate the creation of consent for the Domestic Credit Transfer operation.

Image 10: Domestic Credit Transfer – Submit consent



To initiate a Domestic Credit Transfer, the following actions are performed:

1. The Payment Services User (PSU) populates payment details and provides consent to the TPP.
2. The TPP submits consent to MAOB Connect Hub using MAOB API.
3. MAOB API translates and routes a consent request to the appropriate Connectivity Partner.
4. The Connectivity Partner translates and routes the consent request to the appropriate ASPSP.
5. The ASPSP stores the consent request and returns the URL for the PSU authentication to the Connectivity Partner.
6. The Connectivity Partner returns URL for PSU authentication to MAOB API.
7. The MAOB API returns URL for PSU authentication to the TPP.
8. The TPP redirects the PSU to the URL for PSU authentication.
9. The PSU authenticates with the ASPSP and authorizes consent.
10. The ASPSP returns the authorization string and state to the specified URL.

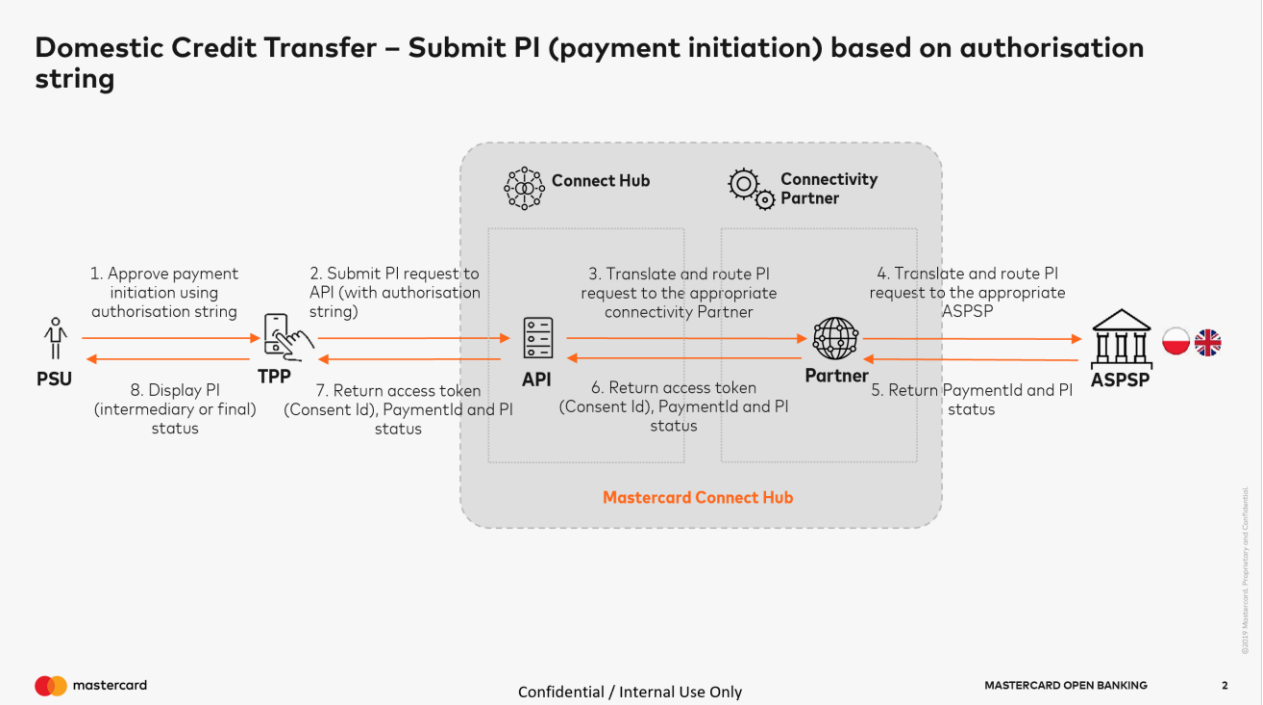
4.11 PIS - Domestic Credit Transfer – Submit Payment Initiation

The following Business Context Diagram shows the flows used to initiate a Domestic payment on the ASPSP side using an authorization string provided by the PSU.

Prerequisites: PSU should perform the following operations before the 'Submit Payment Initiation' call is executed:

1. 'Submit Consent' operation to get authorization string

Image 11: Domestic Credit Transfer – Submit Payment Initiation



The following actions are performed to initiate a Domestic payment:

1. The PSU approves a Payment Initiation (PI) and send authorization string provided by the ASPSP.
2. The TPP submits the PI request with authorization string to MAOB Connect Hub using MAOB API.
3. MAOB API translates and routes the request to the appropriate Connectivity Partner.
4. The Connectivity Partner translates and routes the request to the appropriate ASPSP.
5. The ASPSP returns payment identification and PI status to the Connectivity Partner.
6. The Connectivity Partner exchanges authorization string for access token and returns access token (Consent Id), payment identification (Payment Id) and PI status MAOB API.
7. MAOB API returns access token (Consent Id), payment identification (Payment Id) and PI status to the TPP.
8. The TPP displays the PI status to the PSU.

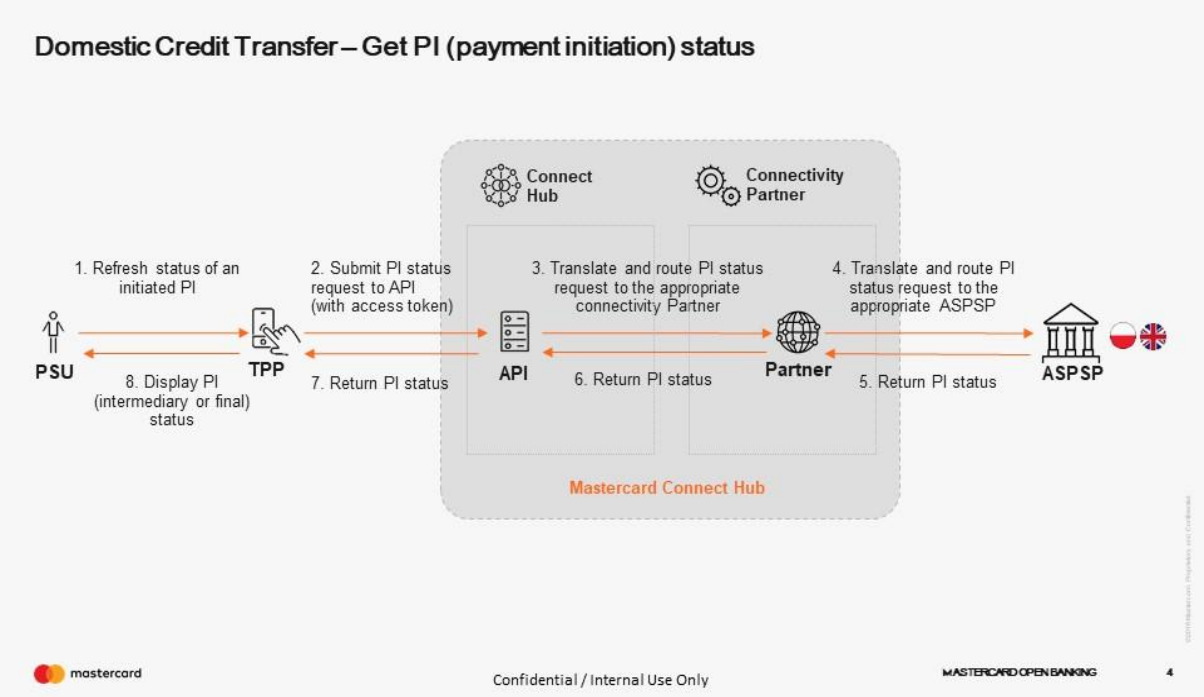
4.12 PIS - Domestic Credit Transfer – Get Payment Status

The following Business Context Diagram shows the flows used to retrieve the payment status from the ASPSP for the specified payment based on previously obtained consent.

Prerequisites: PSU should perform the following operations before the 'Get Payment Status' call is executed:

1. 'Submit Consent' operation to get authorization string
2. 'Submit Payment Initiation' operation to get access token

Image 12: Domestic Credit Transfer – Get Payment Initiation Status



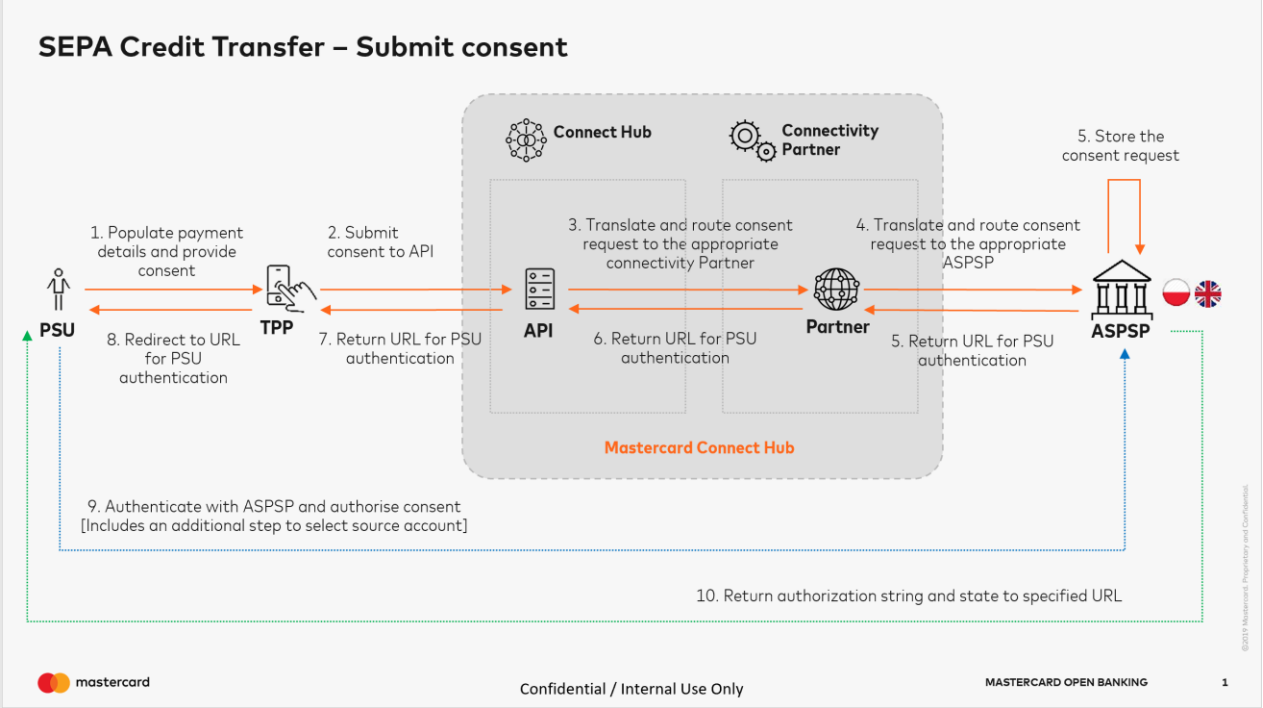
To perform a Get Payment Initiation Status request, the following actions are performed:

1. The PSU sends a payment status request to the TPP.
2. The TPP submits the PI status request with previously obtained access token (Consent Id) to MAOB Connect Hub using MAOB API.
3. MAOB API translates and routes the request to the appropriate Connectivity Partner.
4. The Connectivity Partner translates and routes the request to the appropriate ASPSP.
5. The ASPSP returns PI status to Connectivity Partner.
6. The Connectivity Partner returns the PI status MAOB API.
7. MAOB API returns the PI status to the TPP.
8. The TPP displays the PI status to PSU.

4.13 PIS - SEPA Credit Transfer – Submit Consent

The following Business Context Diagram shows the flows used to initiate the creation of consent for the SEPA Credit Transfer operation.

Image 13: SEPA Credit Transfer – Submit consent



To initiate a SEPA Credit Transfer, the following actions are performed:

1. The Payment Services User (PSU) populates payment details and provides consent to the TPP.
2. The TPP submits consent to MAOB Connect Hub using MAOB API.
3. MAOB API translates and routes a consent request to the appropriate Connectivity Partner.
4. The Connectivity Partner translates and routes the consent request to the appropriate ASPSP.
5. The ASPSP stores the consent request and returns the URL for the PSU authentication to the Connectivity Partner.
6. The Connectivity Partner returns URL for PSU authentication to MAOB API.
7. The MAOB API returns URL for PSU authentication to the TPP.
8. The TPP redirects the PSU to the URL for PSU authentication.
9. The PSU authenticates with the ASPSP and authorizes consent.
10. The ASPSP returns the authorization string and state to the specified URL.

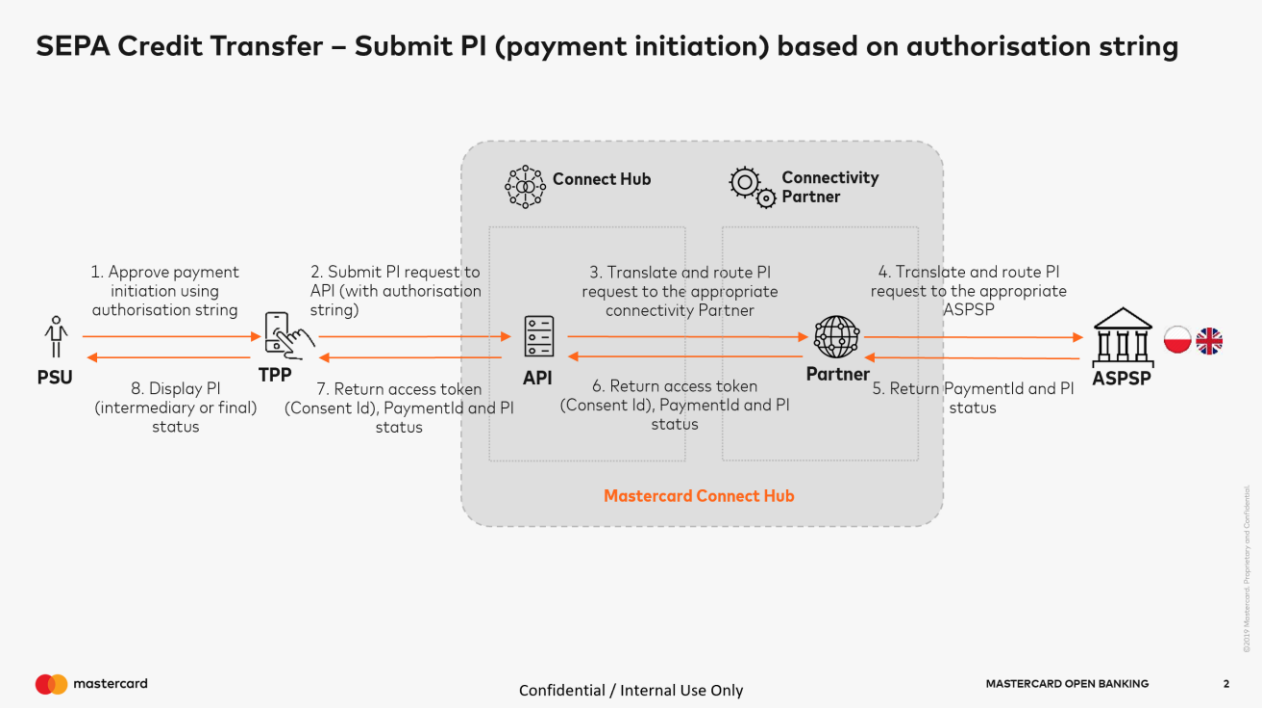
4.14 PIS - SEPA Credit Transfer – Submit Payment Initiation

The following Business Context Diagram shows the flows used to initiate a SEPA payment on the ASPSP side using an authorization string provided by the PSU.

Pre-requests: PSU should perform the following operations before the 'Submit Payment Initiation' call is executed:

1. 'Submit Consent' operation to get authorization string

Image 14: SEPA Credit Transfer – Submit Payment Initiation



The following actions are performed to initiate a SEPA payment:

1. The PSU approves a Payment Initiation (PI) and send authorization string provided by the ASPSP.
2. The TPP submits the PI request with authorization string to MAOB Connect Hub using MAOB API.
3. MAOB API translates and routes the request to the appropriate Connectivity Partner.
4. The Connectivity Partner translates and routes the request to the appropriate ASPSP.
5. The ASPSP returns payment identification and PI status to the Connectivity Partner.
6. The Connectivity Partner exchanges authorization string for access token and returns access token (Consent Id), payment identification (Payment Id) and PI status MAOB API.
7. MAOB API returns access token (Consent Id), payment identification (Payment Id) and PI status to the TPP.
8. The TPP displays the PI status to the PSU.

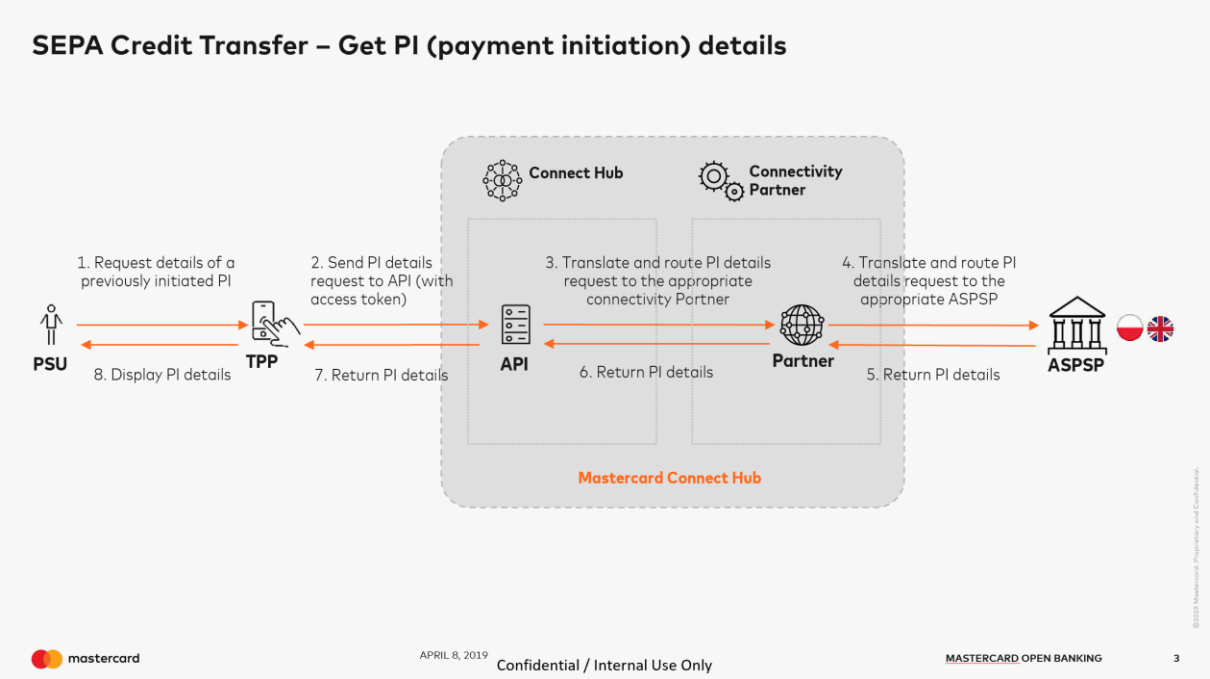
4.15 PIS - SEPA Credit Transfer – Get Payment Status

The following Business Context Diagram shows the flows used to retrieve the payment status from the ASPSP for the specified payment based on previously obtained consent.

Pre-requests: PSU should perform the following operations before the 'Get Payment Status' call is executed:

1. 'Submit Consent' operation to get authorization string
2. 'Submit Payment Initiation' operation to get access token

Image 15: SEPA Credit Transfer – Get Payment Initiation Status



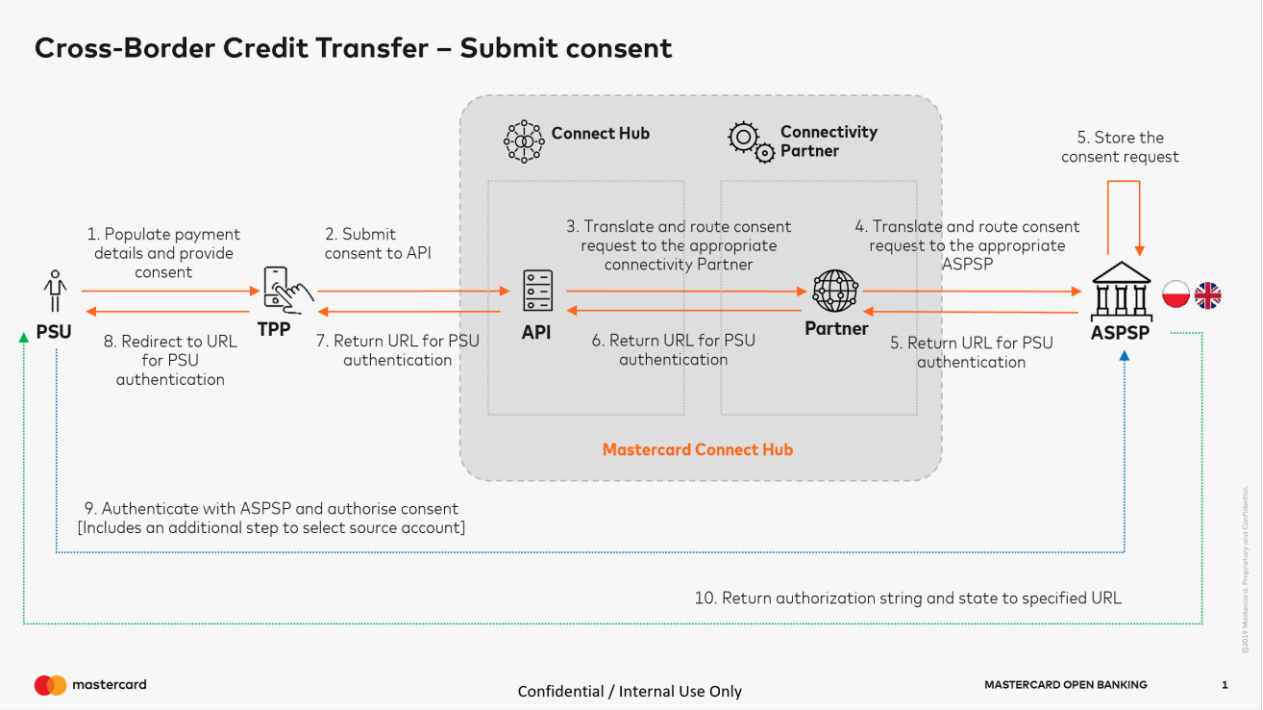
To perform a Get Payment Initiation Status request, the following actions are performed:

1. The PSU sends a payment status request to the TPP.
2. The TPP submits the PI status request with previously obtained access token (Consent Id) to MAOB Connect Hub using MAOB API.
3. MAOB API translates and routes the request to the appropriate Connectivity Partner.
4. The Connectivity Partner translates and routes the request to the appropriate ASPSP.
5. The ASPSP returns PI status to Connectivity Partner.
6. The Connectivity Partner returns the PI status MAOB API.
7. MAOB API returns the PI status to the TPP.
8. The TPP displays the PI status to PSU.

4.16 PIS – Cross-Border Credit Transfer – Submit Consent

The following Business Context Diagram shows the flows used to initiate the creation of consent for the Cross-Border Credit Transfer operation.

Image 16: Cross-Border Credit Transfer – Submit consent



To initiate a Cross-Border Credit Transfer, the following actions are performed:

1. The Payment Services User (PSU) populates payment details and provides consent to the TPP.
2. The TPP submits consent to MAOB Connect Hub using MAOB API.
3. MAOB API translates and routes a consent request to the appropriate Connectivity Partner.
4. The Connectivity Partner translates and routes the consent request to the appropriate ASPSP.
5. The ASPSP stores the consent request and returns the URL for the PSU authentication to the Connectivity Partner.
6. The Connectivity Partner returns URL for PSU authentication to MAOB API.
7. The MAOB API returns URL for PSU authentication to the TPP.
8. The TPP redirects the PSU to the URL for PSU authentication.
9. The PSU authenticates with the ASPSP and authorizes consent.
10. The ASPSP returns the authorization string and state to the specified URL.

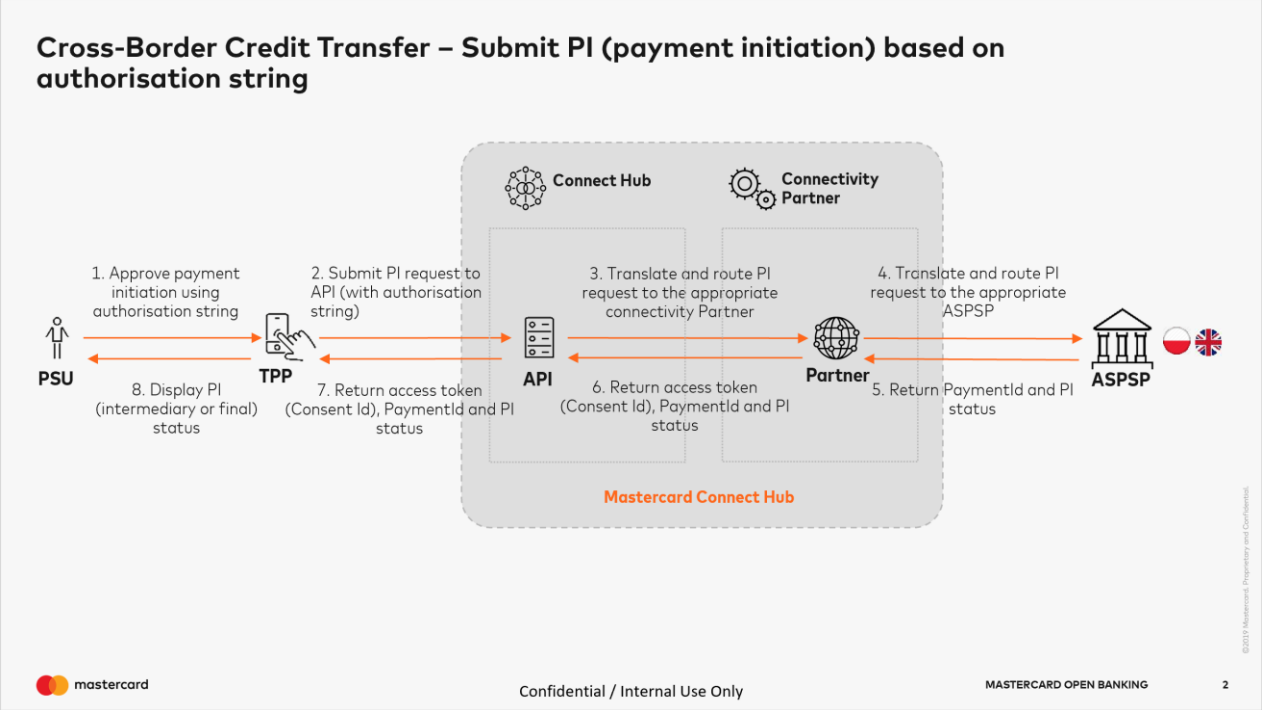
4.17 PIS - Cross-Border Credit Transfer – Submit Payment Initiation

The following Business Context Diagram shows the flows used to initiate a Cross-Border payment on the ASPSP side using an authorization string provided by the PSU.

Pre-requests: PSU should perform the following operations before the 'Submit Payment Initiation' call is executed:

1. 'Submit Consent' operation to get authorization string

Image 17: Cross-Border Credit Transfer – Submit Payment Initiation



The following actions are performed to initiate a Cross-Border payment:

1. The PSU approves a Payment Initiation (PI) and send authorization string provided by the ASPSP.
2. The TPP submits the PI request with authorization string to MAOB Connect Hub using MAOB API.
3. MAOB API translates and routes the request to the appropriate Connectivity Partner.
4. The Connectivity Partner translates and routes the request to the appropriate ASPSP.
5. The ASPSP returns payment identification and PI status to the Connectivity Partner.
6. The Connectivity Partner exchanges authorization string for access token and returns access token (Consent Id), payment identification (Payment Id) and PI status MAOB API.
7. MAOB API returns access token (Consent Id), payment identification (Payment Id) and PI status to the TPP.
8. The TPP displays the PI status to the PSU.

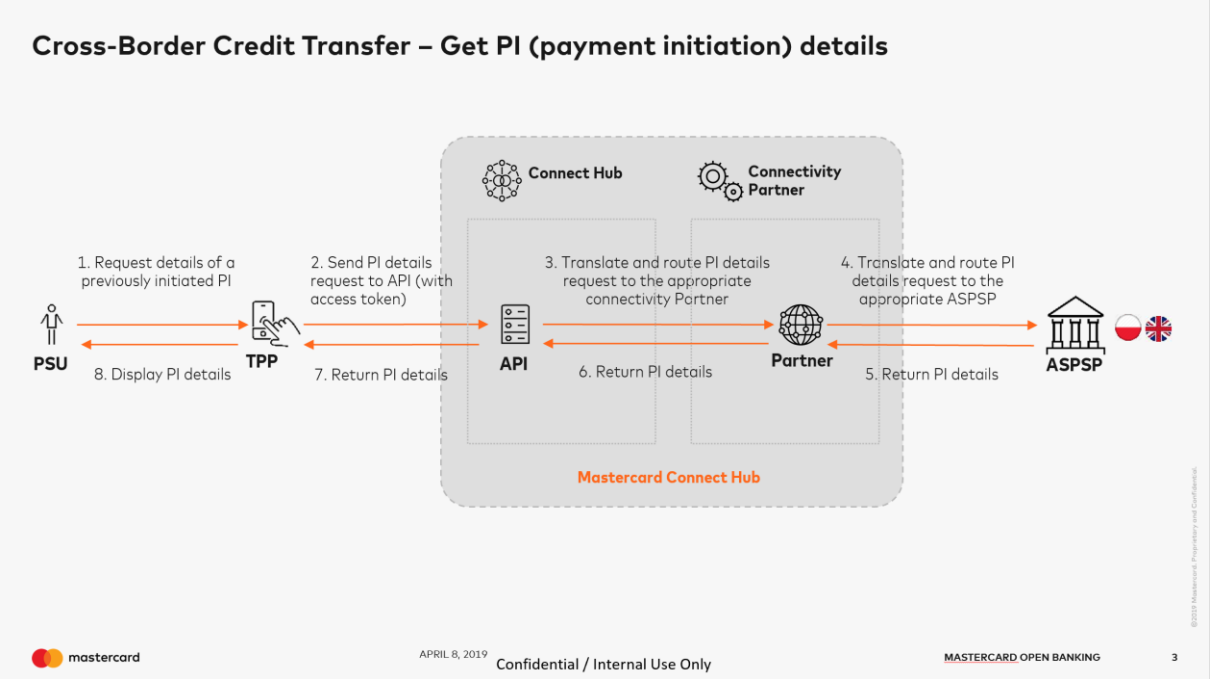
4.18 PIS - Cross-Border Credit Transfer – Get Payment Status

The following Business Context Diagram shows the flows used to retrieve the payment status from the ASPSP for the specified payment based on previously obtained consent.

Pre-requests: PSU should perform the following operations before the 'Get Payment Status' call is executed:

1. 'Submit Consent' operation to get authorization string
2. 'Submit Payment Initiation' operation to get access token

Image 18: Cross-Border Credit Transfer – Get Payment Initiation Status



To perform a Get Payment Initiation Status request, the following actions are performed:

1. The PSU sends a payment status request to the TPP.
2. The TPP submits the PI status request with previously obtained access token (Consent Id) to MAOB Connect Hub using MAOB API.
3. MAOB API translates and routes the request to the appropriate Connectivity Partner.
4. The Connectivity Partner translates and routes the request to the appropriate ASPSP.
5. The ASPSP returns PI status to Connectivity Partner.
6. The Connectivity Partner returns the PI status MAOB API.
7. MAOB API returns the PI status to the TPP.
8. The TPP displays the PI status to PSU.

5 API Access Methods

The following tables provide an overview of the HTTP access methods supported by the API endpoints and by the resources created through this API.

5.1 Accounts endpoints

Table 2: Accounts Endpoints

Endpoints/Resources	Method	API Profiles	Description
/accounts/aspsps	POST	N/A	Reads list of supported ASPSPs
/accounts/health	GET	N/A	Reads Inbound API and Connectors statuses of Accounts Service
/accounts	POST	CMA9, Polish API, NextGenPSD2, STET	Reads all identifiers of the accounts, to which a PSU has granted an account access to through the /accounts/consents endpoint.
/accounts/consents	POST	CMA9, Polish API, NextGenPSD2, STET	Requests AI consent initiation.
/accounts/consents/raw	POST	CMA9	Reads original raw consent given by the ASPSP
/accounts/consents/authorizations	POST	N/A	Exchanges the PSU authorization string for account access consent
/accounts/consents/delete	POST	CMA9, Polish API, NextGenPSD2, STET	Deletes an existing consent.
/accounts/account	POST	CMA9, Polish API, NextGenPSD2, STET	Reads details about an account. It is assumed, that consent of the PSU to this access is already given and stored on the ASPSP system.
/accounts/account/balances	POST	CMA9, Polish API, NextGenPSD2, STET	Reads account balances. It is assumed that consent of the PSU to this access is already given and stored on the ASPSP system.
/accounts/account/transactions	POST	CMA9, Polish API, NextGenPSD2, STET	Reads transaction list of the account. It is assumed that consent of the PSU to this access is already given and stored on the ASPSP system.
/accounts/account/transactions/transaction	POST	Polish API, NextGenPSD2, STET	Reads transaction details for a given transaction on a given account. It is assumed that consent of the PSU to this access is already given and stored on the ASPSP system.

5.2 Payments endpoints

Table 3: Payments Endpoints

Endpoints/Resources	Method	API Profiles	Description
/payments/aspsps/	POST	N/A	Reads list of supported ASPSPs
/payments/health	GET	N/A	Reads Inbound API and Connectors statuses of Payment Service
/payments/consents/raw	POST	CMA9	Reads original raw consent given by the ASPSP
/payments/domestic-credit-transfers	POST	CMA9, Polish API	Creates a payment initiation request at the ASPSP.
/payments/domestic-credit-transfers/consents	POST	CMA9, Polish API	Requests PI consent initiation.
/payments/domestic-credit-transfers/payment-status	POST	CMA9, Polish API	Reads payment status. It is assumed that a consent of the PSU to this access is already given and stored on the ASPSP system
/payments/sepa-credit-transfers	POST	Polish API, NextGenPSD2, STET	Creates a payment initiation request at the ASPSP.
/payments/sepa-credit-transfers/consents	POST	Polish API, NextGenPSD2, STET	Requests PI consent initiation.
/payments/sepa-credit-transfers/payment-status	POST	Polish API, NextGenPSD2, STET	Reads payment status. It is assumed that a consent of the PSU to this access is already given and stored on the ASPSP system
/payments/cross-border-credit-transfers	POST	CMA9, Polish API	Creates a payment initiation request at the ASPSP.
/payments/cross-border-credit-transfers/consents	POST	CMA9, Polish API	Requests PI consent initiation.
/payments/cross-border-credit-transfers/payment-status	POST	CMA9, Polish API	Reads payment status. It is assumed that a consent of the PSU to this access is already given and stored on the ASPSP system.

6 Response codes and Error Handling

6.1 *HTTP response codes*

The HTTP response code communicates the success or failure of a TPP request message.

Table 4: HTTP Response Codes

Response Code	Description
200 OK	The request has succeeded.
400 Bad Request	The request failed due to an error with the information provided. The TPP should not repeat the request without modifications.
500 Error	The request failed due to an internal error in Open Banking Connect API.
504 Gateway Timeout	The server did not receive a timely response from the data provider

6.2 *Error information*

If OB Connect returns a HTTP response code = 400, the response body typically provides additional details on the error. The table below describes the structure of errors in such a scenario:

Table 5 HTTP Error 400 Error Additional Details Structure

Name	Condition	Multiplicity	Type	Description
Errors	M	1..1	Object	This section can include messages to the TPP on operational issues (errors)
++ Error	M	1..*	Object	Error object, e.g. include one instance per each schema validation error
++++ Source	O	0..1	String	The application that generated this error. For Open Banking Connect API, this will usually be returned as 'com.mastercard.mcob.apiserver'.
++++ ReasonCode	O	0..1	String	The code of the error. Expect the following Codes as of this release: • More Error codes and description in Section 6.3
++++ Description	O	0..1	String	In some scenarios includes description of the error. Note: This is also known as the Error Message.
++++ Recoverable	O	0..1	Boolean	Indicates whether OB Connect always returns this error for this request, or if retrying could change the outcome. For example, if the request contains an invalid signature, retrying will never result in a success.
++++ Details	O	0..*	String	In some scenarios includes additional error details, e.g. in case of a schema validation failure may include the path to the respective elements in the message that failed the validation. Find more details in the Reason Codes specification below.

6.3 *Error codes*

When an error code is returned, the response body provides reason code and error description of the raised error. The following is a list of possible error reason codes that can be returned and additional information.

6.3.1 HTTP Status 500/504 - General reason codes

List of reason codes returned for the majority types of requests together with HTTP status code 500 Internal Server Error, or 504 Gateway Timeout:

Table 6 HTTP Status 500/504 - General reason codes

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
"Unable to process request this time."	INTERNAL_ERROR	This is returned when HTTP Status Code = 500	n/a	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "INTERNAL_ERROR", "Description": "Unable to process request this time" }] } }	Not a typical error. If an unexpected internal error happens, it means additional investigation may be required to identify the issue.	Contact Open Banking Connect API support team.
"Unable to process request this time."	TIMEOUT	This is returned when HTTP Status Code = 504	n/a	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "TIMEOUT", "Recoverable": true }] } }	This usually occurs when server, while acting as a gateway or proxy, did not receive a timely response from the upstream server, e.g. the connectivity partner or the ASPSP. Exceptions: health and aspsps endpoints will not raise this error because they do not use an upstream server.	Re-try sending the same request after a recommended period of time.

6.3.2 HTTP Status 400 - General reason codes

List of reason codes returned for the majority types of requests together with HTTP status code 400 Bad request:

Table 7 HTTP Status 400 - General reason codes

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
The specific message depends on the JSON validators listed here (link to validators).	FORMAT_ERROR	The request that was sent to the endpoint does not comply with the message schema (format) published for that endpoint.	"path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation.	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": "[Path '/limit'] Numeric instance is lower than the required minimum (minimum: 1, found: 0)", "Details": "path[0]=/limit" }] } }	Typical occurrences are: • Missing mandatory fields • Providing a value of the wrong type • Exceeding the minimum/maximum values, • Providing a value that doesn't match the expected pattern • Missing to provide a conditionally expected value when required	Correct the request message schema based on the received error message and then re-send the request.
Inactive Account	INACTIVE_ACCOUNT	The source TPP (i.e. clientId) that has sent the message is not recognized or is invalid.	n/a	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "INACTIVE_ACCOUNT" }] } }	Typically occurs when the TPP onboarding has not completed or TPP used incorrect ClientID value. Note. ClientID is equivalent of the ConsumerKey Id in Mastercard portal	Contact Open Banking Connect API support team.

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
Inactive Provider	INACTIVE_PROVIDER	The destination ASPSP (i.e. aspspId) is not recognized or is invalid.	n/a	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "INACTIVE_PROVIDER" }] } }	Typically occurs when aspspId value is incorrect or ASPSP was disabled.	Check that the value used for aspspId in the request is valid as per the list of ASPSPs returned by Open Banking Connect API (e.g. in /payments/aspsps or /accounts/aspsps). If you think you should be able to access the specified ASPSP, contact the Open Banking Connect API support team.
"You are not allowed to perform request to this ASPSP"	NOT_ALLOWED	TPP is not allowed to submit requests to this ASPSP. The availability of an ASPSP for a TPP is recorded in Open Banking Connect based on TPP registration Data. Only available ASPSPs are returned to the TPP from the Open Banking Connect API.	Return as path to the element that failed the validation.	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "NOT_ALLOWED", "Description": "You are not allowed to perform this request to this ASPSP" }] } }	Typically happens when TPP does not register correctly.	Check that the value used for aspspId in the request is valid as per the list of ASPSPs returned by Open Banking Connect API (e.g. in /payments/aspsps or /accounts/aspsps). If you think you should be able to access the specified ASPSP, contact the Open Banking Connect API support team.

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
"Method not supported by Provider's API profile"	NOT_FOUND	Method not supported by ASPSP. Depending on the API Standard /Profile implemented at the side of the ASPSP, some of the endpoints may not be available for that ASPSP. The information about API Profile support is provided in the specification for each endpoint.	n/a	{ "Errors": { "Error": [{ "ReasonCode": "NOT_FOUND", "Description": "Method not supported by Provider's API profile" }] } }	Typically happens with new customers who send data not part of the API Standard.	Verify the specification and ensure that the endpoint the ASPSP supports the API Profile returned (e.g. in /payments/aspsps or /accounts/aspsps). If you think you should be able to access the specified endpoint, contact the Open Banking Connect API support team.
"Unable to process request this time."	PROVIDER_ERROR	There was an error returned from the ASPSP or connectivity partners.	n/a	{ "Errors": { "Error": [{ "ReasonCode": "PROVIDER_ERROR", "Description": "Unable to process request this time" }] } }	This typically occurs due to. incorrect parameters sent to connectivity partners or ASPSPs.	Contact Open Banking Connect API support team.

6.3.3 HTTP Status 400 - AIS - Account Information Consent requests - Specific reason codes

List of reason codes returned in the context of submitting an Account Information consent request (POST /accounts/consents):

Table 8 HTTP Status 400 - AIS - Account Information Consent requests - Specific reason codes

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps

n/a	INACTIVE_URL	The value provided for the tppRedirectURI parameter is invalid. Only URIs that have been registered with OBIE can be used by TPP.	n/a	<pre>{
				"Errors": {
				"Error": [
				{
				"Source": "com.mastercard.mcob.apiserver",
				"ReasonCode": "INACTIVE_URL"
				}
				]
				}
				}</pre>
					Typically occurs because the TPP has not registered the URI with OBIE during onboarding.	Make sure that in order to populate the value of the tppRedirectURI field, you are using a URI that was registered with Open Banking Connect API as part of the onboarding details.
						If you think you should be able to use the specified value, contact the Open Banking Connect API support team.

6.3.4 HTTP Status 400 - AIS – Get List of Accounts – Specific reason codes

List of reason codes returned in the context of submitting a Get List of Accounts request (POST /accounts):

Table 9 HTTP Status 400 - AIS – Get List of Accounts - Specific reason codes

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
"Conditional field psuIPAddress is expected when isLivePsuRequest='true'"	FORMAT_ERROR	"IP address" field is mandatory if parameter "isLivePsuRequest"=true.	"path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": Conditional field psuIPAddress is expected when isLivePsuRequest='true'", "Details": "path[0]=/requestInfo/psuIPAddress" }] } }	Typically occurs because the TPP has provided in request "isLivePsuRequest"=true and didn't provide the IP address value.	Provide IP address of PSU in the request when providing "isLivePsuRequest"=true.

6.3.5 HTTP Status 400 - AIS – Exchange the PSU authorization for access consent requests - Specific reason codes

List of reason codes that will only be returned in the context of submitting an Exchange the PSU authorization request (POST /accounts/consents/authorizations):

Table 10 HTTP Status 400 - AIS – Exchange the PSU authorization for access consent requests - Specific reason codes

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
"ASPSP's signature didn't pass validation. Result code: %s"	INVALID_SIGNATURE	The signature of ASPSP that was provided in the authorization code of the request couldn't be validated.	"path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "INVALID_SIGNATURE", "Description": "ASPSP's signature didn't pass validation. Result code: expired", "Details": "path[0]=/requestInfo/authorization" }] } }	Typically occurs because the signature of the ASPSP in the authorization string didn't pass one or multiple validations. This can be as a result of a modified redirect URL by the PSU, and expired signature, or other reasons.	Your request didn't get sent to ASPSP since the signature of the ASPSP in authorization code failed validation. Next steps depend on the result code in error message: • "revoked" – The ASPSP revoked the signature certificate used. • "expired" - signature certificate has expired.Reinitiate payment. • "unknown" - signature certificate couldn't be identified. Reinitiate payment or contact support for more details. • "unsigned" - signature certificate is missing. Reinitiate payment or contact support for more details. • "noncompliant" - signature certificate is of an unexpected format. Contact support for more details. If you think you should be able to use the specified value, contact the Open Banking Connect API support team.

6.3.6 HTTP Status 400 - AIS – Get Account Details – Specific reason codes

List of reason codes returned in the context of submitting a Get Account Details request (POST /accounts/account):

Table 11 HTTP Status 400 - AIS – Get Account Details - Specific reason codes

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
"Conditional field psuIPAddress is expected when isLivePsuRequest='true'"	FORMAT_ERROR	"IP address" field is mandatory if parameter "isLivePsuRequest"=true.	"path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": ""Conditional field psuIPAddress is expected when isLivePsuRequest='true'", "Details": "path[0]=/requestInfo/psuIPAddress" }] } }	Typically occurs because the TPP has provided in request "isLivePsuRequest"=true and didn't provide the IP address value.	Provide IP address of PSU in the request when providing "isLivePsuRequest"=true.

6.3.7 HTTP Status 400 - AIS – Get Account Balances – Specific reason codes

List of reason codes that returned in the context of submitting a Get Account Balances request (POST /accounts/account/balances):

Table 12 HTTP Status 400 - AIS – Get Account Balances - Specific reason codes

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
"Conditional field psuIPAddress is expected when isLivePsuRequest='true'"	FORMAT_ERROR	"IP address" field is mandatory if parameter "isLivePsuRequest"=true.	"path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": " Conditional field psuIPAddress is expected when isLivePsuRequest='true'", "Details": "path[0]=/requestInfo/psuIPAddress" }] } }	Typically occurs because the TPP has provided in request "isLivePsuRequest"=true and didn't provide the IP address value.	Provide IP address of PSU in the request when providing "isLivePsuRequest"=true.

6.3.8 HTTP Status 400 - AIS - Get Account Transactions - Specific reason codes

List of reason codes returned in the context of an Account Information Service request. All the errors below can only be returned in context of requesting the list of transactions for an account (i.e. POST /accounts/account/transactions):

Table 13 HTTP Status 400 - AIS - Get Account Transactions - Specific reason codes

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
"[Path 'path[i]'] Numeric instance is lower than the required minimum (minimum: <MIN>, found: <received value>)"	FORMAT_ERROR	The value for the limit parameter in the Get Account Transactions request is smaller than the specified minimum. Note: This error is only applicable to the following endpoints: - - POST /accounts/account/transactions.	"path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": "[Path '/limit'] Numeric instance is lower than the required minimum (minimum: 1, found: 0)", "Details": "path[0]=/limit" }] } }	Typically occurs when the limit parameter field has a value of zero.	Ensure that the value for the limit parameter in the request is greater than or equal to 1
"Numeric instance is greater than the accepted maximum (maximum: <MAX>, found: <received value>)"	MAX_LIMIT_FORMAT_ERROR	The value for the limit parameter in the Get Account Transactions request is greater than the specified maximum.	"path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "MAX_LIMIT_FORMAT_ERROR", "Description": "Numeric instance is greater than the required maximum (maximum: 1000, found: 1001)" }] } }	Typically occurs when the maximum has been set too high.	Ensure that the value for the limit parameter in the request is lower than the maximum value specified in the message

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
"bookingDateFrom cannot be greater than bookingDateTo"	DATEFROM_GREATER_THAN_DATETO	The value for the bookingDateFrom parameter in the Get Account Transactions request is greater than the value of bookingDateTo.	"path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "DATEFROM_GREATER_THAN_DATETO", "Description": "bookingDateFrom cannot be greater than bookingDateTo" }] } }	This typically is a user entry error. Dates should identify a time interval greater than zero.	Ensure that the value specified for bookingDateFrom parameter, doesn't exceed the value specified for bookingDateTo parameter
"bookingDateFrom and/or bookingDateTo should be a valid date or cannot be a date from the future"	INVALID_DATEFROM_OR_DATETO	The value for the bookingDateFrom or bookingDateTo parameters is not a valid date. Or, the value for the bookingDateFrom or bookingDateTo parameters is a date from the future.	"path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "INVALID_DATEFROM_OR_DATETO", "Description": "bookingDateFrom and/or bookingDateTo should be a valid date or cannot be a date from the future" }] } }	This typically happens because of submitting user's error.	Ensure that the value specified for both bookingDateFrom and bookingDateTo parameter is a valid date and not in future

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
"[Path 'path[i]'] String %s is invalid against requested date format(s) yyyy-MM-dd"Date format (e.g. 2019-10-25)"	FORMAT_ERROR	The value for the bookigDateFrom or bookingDateTo parameters is not in ISO date format, i.e. YYYY-MM-DD.	"path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": "[Path '/bookingDateFrom'] String \"2018-35-22\" is invalid against requested date format(s) yyyy-MM-dd", "Details": "path[0]=/bookingDateFrom" }, { "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": "[Path '/bookingDateTo'] String \"2018-11-31\" is invalid against requested date format(s) yyyy-MM-dd", "Details": "path[0]=/bookingDateTo" }] } }	Typically, errors occur with dates set to MM-DD-YYYY instead of YYYY-MM-DD.	Ensure that the value specified for both bookingDateFrom and bookingDateTo parameter complies to the following format: YYYY-MM-DD

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
"Conditional field psuIPAddress is expected when isLivePsuRequest='true'"	FORMAT_ERROR	"IP address" field is mandatory if parameter "isLivePsuRequest"=true.	"path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": " Conditional field psuIPAddress is expected when isLivePsuRequest='true'", "Details": "path[0]=/requestInfo/psuIPAddress" }] } }	Typically occurs because the TPP has provided in request "isLivePsuRequest" =true and didn't provide the IP address value.	Provide IP address of PSU in the request when providing "isLivePsuRequest "=true.

6.3.9 HTTP Status 400 - AIS – Get Transaction Details – Specific reason codes

List of reason codes that returned in the context of submitting a Get Transaction Details request (POST /accounts/account/transactions/transaction):

Table 14 HTTP Status 400 - AIS – Get Transaction Details - Specific reason codes

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
"Conditional field psuIPAddress is expected when isLivePsuRequest='true' "	FORMAT_ERROR	"IP address" field is mandatory if parameter "isLivePsuRequest"=true.	"path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": " Conditional field psuIPAddress is expected when isLivePsuRequest='true' ", "Details": "path[0]=/requestInfo/psuIPAddress" }] } }	Typically occurs because the TPP has provided in request "isLivePsuRequest" =true and didn't provide the IP address value.	Provide IP address of PSU in the request when providing "isLivePsuRequest"=true .

6.3.10 HTTP Status 400 - PIS - Domestic Payment Initiation Consent request - Specific reason codes

List of reason codes returned in the context of submitting a Domestic Payment Initiation Consent Request (i.e. POST /payments/domestic-credit-transfers/consents):

Table 15 HTTP Status 400 - PIS - Domestic Payment Initiation Consent request - Specific reason codes

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
n/a	INACTIVE_URL	The value provided for the tppRedirectURI parameter is invalid. Only URIs that have been registered with OBIE can be used by TPP.	n/a	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "INACTIVE_URL" }] } }	Typically occurs when the URI has a typo or was incorrectly copied.	Make sure that in order to populate the value of the tppRedirectURI field, you are using a URI that was registered as whitelisting entity with Open Banking Connect API as part of the onboarding details. If you think you should be able to use the specified value, contact the Open Banking Connect API support team.
"[Path 'path[i]'] Numeric instance is lower than the required minimum (minimum: 0, found: %s)"	INVALID_PAYMENT_AMOUNT	The value provided for the instructedAmount is less or equal to zero	Return as path to the element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": "[Path '/payments/instructedAmount/amount'] Numeric instance is lower than the required minimum (minimum: 0, found: -2)", "Details": "path[0]=/payments/instructedAmount/amount" }] } }	Typically occurs due to the submitting user's error.	Ensure that a value greater than 0 is populated as instructedAmount and re-submit the message Standard amount format should be used.

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
"Execution Date cannot be in the past".	FORMAT_ERROR	The value provided for the requestedExecutionDate is in the past. This is an invalid scenario as this field is usually used to request a future dated payment	path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": "Execution Date cannot be in the past", "Details": "path[0]=/payments/requestedExecutionDate" }] } }	Typically occurs due to submitting user's error.	Ensure that a value populated in requestedExecutionDate is a present or future date and re-submit the message MAOB API has no limit for future date, but it could be on ASPSP side.
"End date is lower than start date."	STARTDATE_GREATER_THAN_ENDDATE	The value provided for the startDate is greater than endDate. This is invalid scenario as the selected dates should define the start and end dates of the Settlement Order's payment schedule.	Return as path to the element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "STARTDATE_GREATER_THAN_ENDDATE", "Description": "End date is lower than start date" }] } }	This error typically occurs due to submitting user's error.	Ensure that a value populated in startDate is lower than or equal the endDate and re-submit the message

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
"startDate and/or endDate cannot be a date from the past."	INVALID_STARTDATE_OR_ENDDATE	The value provided for the startDate or endDate is in the past. This is invalid scenario as the selected dates should define the start and end dates of the Settlement Order's payment schedule.	Return as path to the element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "STARTDATE_GREATER_THAN_ENDDATE", "Description": "End date is lower than start date" }] } }	This error typically occurs due to submitting user's error.	Ensure that a value populated in startDate and endDate are present or future dates and re-submit the message

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
"[Path 'path[i]'] String %s is invalid against requested date format(s) yyyy-MM-dd"	FORMAT_ERROR	The value provided for the startDate or endDate does not comply with standard format, e.g. YYYY-MM-DD.	Return as path to the element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": "[Path '/payments/schedule/endDate'] String \"20202-04-18\" is invalid against requested date format(s) yyyy-MM-dd", "Details": "path[0]=/payments/schedule/endDate" }, { "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": "[Path '/payments/schedule/startDate'] String \"20-03-18\" is invalid against requested date format(s) yyyy-MM-dd", "Details": "path[0]=/payments/schedule/startDate" }] } }	This typically occurs with and invalid entry of MM-DD-YYYY.	Ensure that a value populated in startDate and endDate complies to the following format: YYYY-MM-DD

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
"It is unclear what payment is requested - Future Dated or Standing Order. "	UNCLEAR_PAYMENT_INSTRUCTION	The request contains both requestedExecutionDate and schedule elements populated. This is an invalid scenario, as the former defines a future dated payment, and the later defines a Standing Order.	Return as path to the element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "UNCLEAR_PAYMENT_INSTRUCTION", "Description": "It is unclear what payment is requested - Future Dated or Standing Order" }] } }	Typically happens when dates are not populated in the schedule section to initiate a Standing Order payment.	Ensure that only one element is populated, i.e. either requestedExecutionDate in case you want to initiate a Future Dated Payment or populate the necessary fields in the schedule section to initiate a standing order based on defined schedule.
"Method not support by Provider's API profile"	NOT_FOUND	Failed to provide a supported localInstrument in PI Consent.	n/a	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "NOT_FOUND", "Description": "Method not supported by Provider's API profile" }] } }	Typically occurs when TPP does not use a valid localInstrument value supported by their API Standard.	Make sure that you select one of the specified values for localInstrument field based on the specification. Also make sure that you are filling in all the dependent fields according to the specification before re-submitting the message

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
"Invalid currency for selected localInstrument"	FORMAT_ERROR	Failed to provide required currency details for selected localInstrument in PI consent. Consult the API specification for field dependencies.	"path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": "Payment details are not aligned with selected localInstrument", "Details": "path[0]=/payments/instructedAmount/currency" }] } }	Typically occurs when TPP does not fill in dependent fields for their API standard.	Make sure that you are filling in currency field with the corresponding value to the selected localInstrument according to the specification before re-submitting the message.
"Conditional field RemittanceInformationUnstructured is expected for selected localinstrument"	FORMAT_ERROR	Failed to provide RemittanceInformationUnstructured field in the request to an ASPSP with Polish API standard.	"path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": " Conditional field RemittanceInformationUnstructured is expected for selected localinstrument ", "Details": "path[0]=/payments/remittanceInformationUnstructured" }] } }	Typically occurs when TPP does not fill in dependent fields for their API standard, i.e. "RemittanceInformationUnstructured".	Ensure that a value populated in RemittanceInformationUnstructured is present and re-submit the message.

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
"Conditional field creditorAddress is expected for selected localinstrument"	FORMAT_ERROR	Failed to provide creditorAddress object in the request to an ASPSP with Polish API standard.	"path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": "Conditional field creditorAddress is expected for selected localinstrument", "Details": "path[0]=/payments/creditorAddress" }] } }	Typically occurs when TPP does not fill in dependent fields for their API standard, i.e. "creditorAddress" object.	Ensure that all needed values populated in creditorAddress object are present and re-submit the message.
"Usage of schemeName does not match selected localInstrument"	FORMAT_ERROR	Failed to provide the schemeName value that corresponds to the localInstrument value in request.	"path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": "Usage of schemeName does not match selected localInstrument", "Details": "path[0]=/payments/creditorAccount/schemeName" }] } }	Typically occurs when TPP provides schemeName value 'UK.AccountNumber' for localInstrument= 'PL.Elixir' or schemeName value 'PL.AccountNumber' for localInstrument= 'UK.FasterPayments'.	Ensure that schemeName value populated in creditorAccount object corresponds to localInstrument value and re-submit the message.

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
"Usage of creditorAgent does not match selected localInstrument"	FORMAT_ERROR	Do not provide creditorAgent in the request with localInstrument = "PL.Elixir".	"path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": "Usage of creditorAgent does not match selected localInstrument", "Details": "path[0]=/payments/creditorAgent" }] } }	Typically occurs when TPP provides creditorAgent parameter for localInstrument 'PL.Elixir'.	Ensure that creditorAgent parameter is missing for localInstrument= 'PL.Elixir' (as required by Polish API profile) and re-submit the message.
"Payment details are not aligned with selected localInstrument"	FORMAT_ERROR	Failed to provide clearingSystemIdentification field in the request with localInstrument = "UK.FasterPayments".	"path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": "Payment details are not aligned with selected localInstrument", "Details": "path[0]=/payments/creditorAgent" }] } }	Typically occurs when TPP doesn't provide clearingSystemIdentification field in request where localInstrument is 'UK.FasterPayments'.	Ensure that clearingSystemIdentification field is included for localInstrument= 'UK.FasterPayments' (as required by CMA9 API profile) and re-submit the message.

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
"Usage of instructionPriority does not match selected localInstrument"	FORMAT_ERROR	instructionPriority field's use doesn't correspond to API profile specifications.	"path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": " Usage of instructionPriority does not match selected localInstrument", "Details": "path[0]=/payments/instructionPriority" }] } }	Typically occurs when TPP provides instructionPriority field in request where localInstrument is 'UK.FasterPayments' or when doesn't provide instructionPriority field in request where localInstrument is 'PL.Elixir'.	Ensure that instructionPriority field is not included for localInstrument= 'UK.FasterPayments' (as required by CMA9 API profile) or included for localInstrument= 'PL.Elixir' (as required by Polish API profile) and re-submit the message.
"Usage of debtorAccount does not match selected localInstrument"	FORMAT_ERROR	Combination of debtorAccount and debtorAgent fields inclusion in the request doesn't corresponds to API profile specifications.	"path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": "Payment details are not aligned with selected localInstrument", "Details": "path[0]=/payments/debtorAccount/identification }] } }	Typically occurs when TPP for localInstrument= 'UK.FasterPayments' provides only debtorAgent field. Otherwise it occurs when TPP for localInstrument= 'PL.Elixir' provides debtorAgent with or without debtorAccount.	Ensure that for localInstrument= 'UK.FasterPayments' both debtorAccount and debtorAgent are included or not included at all (as required by CMA9 API profile) or in case of localInstrument= 'PL.Elixir' provide only debtorAccount or don't include it at all (as required by Polish API profile) and re-submit the message.

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
"Usage of debtorAgent does not match selected localInstrument"	FORMAT_ERROR	debtorAgent field's inclusion in the request doesn't corresponds to API profile specifications.	"path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": "Payment details are not aligned with selected localInstrument", "Details": "path[0]=/payments/debtorAgent/identification }] } }	Typically occurs when TPP for localInstrument= 'UK.FasterPayments' provides only debtorAccount.	Ensure that for localInstrument= 'UK.FasterPayments' both debtorAccount and debtorAgent are included or not included at all (as required by CMA9 API profile).

6.3.11 HTTP Status 400 - PIS - Domestic Payment Initiation Transfer request - Specific reason codes

No additional list of reason codes is returned in the context of submitting a Domestic Payment Initiation Transfer Request (i.e. POST /payments/domestic-credit-transfers).

Table 16 HTTP Status 400 - PIS – Domestic Payment Initiation Transfer requests - Specific reason codes

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
"ASPSP's signature didn't pass validation. Result code: %s"	INVALID_SIGNATURE	The signature of ASPSP that was provided in the authorization code of the request couldn't be validated.	"path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "INVALID_SIGNATURE", "Description": "ASPSP's signature didn't pass validation. Result code: expired", "Details": "path[0]=/requestInfo/authorization" }] } }	Typically occurs because the signature of the ASPSP in the authorization string didn't pass one or multiple validations. This can be as a result of a modified redirect URL by a PSU, an expired signature, or other reasons.	Your request didn't get sent to ASPSP since the signature of the ASPSP in authorization code failed validation. Next steps depend on the result code in error message: • "revoked" – The ASPSP revoked the signature certificate used. • "expired" - signature certificate has expired. Reinitiate payment. • "unknown" - signature certificate couldn't be identified. Reinitiate payment or contact support for more details. • "unsigned" - signature certificate is missing. Reinitiate payment or contact support for more details. • "noncompliant" - signature certificate is of an unexpected format. Contact support for more details. If you think you should be able to use the specified value, contact the Open Banking Connect API support team.

6.3.12 HTTP Status 400 - PIS - Domestic Payment Status request - Specific reason codes

No additional list of reason codes is returned in the context of submitting a Domestic Payment Status Request (i.e. POST /payments/domestic-credit-transfers/payment-status). Only HTTP Status 400 – General reason codes is returned.

6.3.13 HTTP Status 400 – PIS - SEPA Payment Initiation Consent request - Specific reason codes

List of reason codes returned in the context of submitting a SEPA Payment Initiation Consent Request (i.e. POST /payments/sepa-credit-transfers/consents):

Table 17 HTTP Status 400 - PIS - SEPA Payment Initiation Consent request - Specific reason codes

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
n/a	INACTIVE_URL	The value provided for the tppRedirectURI parameter is invalid. Only URIs that have been registered with OBIE can be used by TPP.	n/a	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "INACTIVE_URL" }] } }	Typically occurs when the URI has a typo or is incorrectly copied.	Make sure that in order to populate the value of the tppRedirectURI field, you are using a URI that was registered with Open Banking Connect API as part of the onboarding details. If you think you should be able to use the specified value, contact the Open Banking Connect API support team.

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
"[Path 'path[i]'] Numeric instance is lower than the required minimum (minimum: 0, found: %s)"	INVALID_PAYMENT_AMOUNT	The value provided for the instructedAmount is less or equal to zero	Return as path to the element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": "[Path '/payments/instructedAmount/amount'] Numeric instance is lower than the required minimum (minimum: 0, found: -2)", "Details": "path[0]=/payments/instructedAmount/amount" }] } }	Typically occurs due to submitting user's error.	Ensure that a value greater than 0 is populated as instructedAmount and re-submit the message
"Execution Date cannot be in the past"	FORMAT_ERROR	The value provided for the requestedExecutionDate is in the past. This is an invalid scenario as this field is usually used to request a future dated payment	path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": "Execution Date cannot be in the past", "Details": "path[0]=/payments/requestedExecutionDate" }] } }	Typically occurs due to submitting user's error.	Ensure that a value populated in requestedExecutionDate is a present or future date and re-submit the message

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
"End date is lower than start date"	STARTDATE_GREATER_THAN_ENDDATE	The value provided for the startDate is greater than endDate. This is invalid scenario as the selected dates should define the start and end dates of the Settlement Order's payment schedule.	Return as path to the element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "STARTDATE_GREATER_THAN_ENDDATE", "Description": "End date is lower than start date" }] } }	Typically occurs because of submitting user's error.	Ensure that a value populated in startDate is lower than or equal the endDate and re-submit the message
"startDate and/or endDate cannot be a date from the past"	INVALID_STARTDATE_OR_ENDDATE	The value provided for the startDate or endDate is in the past. This is invalid scenario as the selected dates should define the start and end dates of the Settlement Order's payment schedule.	Return as path to the element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "INVALID_STARTDATE_OR_ENDDATE", "Description": "startDate and/or endDate cannot be a date from the past" }] } }	Typically occurs because data was delayed and date passes. Especially occurs at next day transition 23:59 to 00:00.	Ensure that a value populated in startDate and endDate are present or future dates and re-submit the message

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
"[Path 'path[i]'] String %s is invalid against requested date format(s) yyyy-MM-dd"	FORMAT_ERROR	The value provided for the startDate or endDate does not comply with standard format, e.g. YYYY-MM-DD.	Return as path to the element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": "[Path '/payments/schedule/endDate'] String \"20202-04-18\" is invalid against requested date format(s) yyyy-MM-dd", "Details": "path[0]=/payments/schedule/endDate" }, { "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": "[Path '/payments/schedule/startDate'] String \"20-03-18\" is invalid against requested date format(s) yyyy-MM-dd", "Details": "path[0]=/payments/schedule/startDate" }] } }	Typically occurs due to date formatted for MM-DD-YYYY.	Ensure that a value populated in startDate and endDate complies to the following format: YYYY-MM-DD

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
"It is unclear what payment is requested - Future Dated or Standing Order"	UNCLEAR_PAYMENT_INSTRUCTION	The request contains both requestedExecutionDate and schedule elements populated. This is an invalid scenario, as the former defines a future dated payment, and the later defines a Standing Order.	Return as path to the element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "UNCLEAR_PAYMENT_INSTRUCTION", "Description": "It is unclear what payment is requested - Future Dated or Standing Order" }] } }	Typically happens when dates are not populated in the schedule section to initiate a Standing Order payment.	Ensure that only one element is populated, i.e. either requestedExecutionDate in case you want to initiate a Future Dated Payment or populate the necessary fields in the schedule section to initiate a standing order based on defined schedule.
"Method not support by Provider's API profile"	NOT_FOUND	Failed to provide a supported localInstrument in PI Consent.	"path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "ReasonCode": "NOT_FOUND", "Description": "Method not supported by Provider's API profile" }] } }	Typically occurs when a TPP does not use a valid localinstrument value used by the API standard.	Make sure that you select one of the specified values for localInstrument field based on the specification. Also make sure that you are filling in all the dependent fields according to the specification before re-submitting the message

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
"Invalid currency for selected localInstrument"	FORMAT_ERROR	Failed to provide required currency details for selected localInstrument in PI consent. Consult the API specification for field dependencies.	"path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": "Payment details are not aligned with selected localInstrument", "Details": "path[0]=/payments/instructedAmount/currency" }] } }	Typically occurs when TPP does not fill in all dependent fields used by their API standard, specifically currency field with the currency value specific to localInstrument.	Make sure that you are filling in the currency field with the corresponding value to the selected localInstrument according to the specification, before re-submitting the message.
"Conditional field RemittanceInformationUnstructured is expected for selected localInstrument"	FORMAT_ERROR	Failed to provide RemittanceInformationUnstructured field in the request to an ASPSP with Polish API standard.	"path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": "Request details are not aligned with API standard", "Details": "path[0]=/payments/remittanceInformationUnstructured" }] } }	Typically occurs when TPP does not fill in dependent fields for their API standard, i.e. "RemittanceInformationUnstructured".	Ensure that a value populated in RemittanceInformationUnstructured is present and re-submit the message.

6.3.14 HTTP Status 400 - PIS- SEPA Payment Initiation Transfer request - Specific reason codes

No additional list of reason codes is returned in the context of submitting a SEPA Payment Initiation Transfer Request (i.e. POST /payments/sepa-credit-transfers). Only HTTP Status 400 – General reason codes is returned.

6.3.15 HTTP Status 400 - PIS- SEPA Payment Status request - Specific reason codes

No additional list of reason codes is returned in the context of submitting a SEPA Payment Status Request (i.e. POST /payments/sepa-credit-transfers/payment-status). Only HTTP Status 400 – General reason codes is returned.

6.3.16 HTTP Status 400 - PIS- Cross-border Payment Initiation Consent request - Specific reason codes

List of reason codes returned in the context of submitting a Cross-border Payment Initiation Consent Request (i.e. POST /payments/cross-border-credit-transfers/consents):

Table 18 HTTP Status 400 - PIS – Cross-border Payment Initiation Consent request - Specific reason codes

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
n/a	INACTIVE_URL	The value provided for the tppRedirectURI parameter is invalid. Only URIs that have been registered with OBIE can be used by TPP.	n/a	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "INACTIVE_URL" }] } }	Typically, this occurs due to a typo when submitting URI or when it is copied incorrectly.	Make sure that in order to populate the value of the tppRedirectURI field, you are using a URI that was registered with Open Banking Connect API as part of the onboarding details. If you think you should be able to use the specified value, contact the Open Banking Connect API support team.

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
"[Path 'path[i]'] Numeric instance is lower than the required minimum (minimum: 0, found: %s)"	INVALID_PAYMENT_AMOUNT	The value provided for the instructedAmount is less or equal to zero	Return as path to the element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": "[Path '/payments/instructedAmount/amount'] Numeric instance is lower than the required minimum (minimum: 0, found: -2)", "Details": "path[0]=/payments/instructedAmount/amount" }] } }	Typically occurs due to submitting user's error.	Ensure that a value greater than 0 is populated as instructedAmount and re-submit the message
"Execution Date cannot be in the past"	FORMAT_ERROR	The value provided for the requestedExecutionDate is in the past. This is an invalid scenario as this field is usually used to request a future dated payment	path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": "Execution Date cannot be in the past", "Details": "path[0]=/payments/requestedExecutionDate" }] } }	Typically occurs due to submitting user's error.	Ensure that a value populated in requestedExecutionDate is a present or future date and re-submit the message

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
"Method not supported by Provider's API profile"	NOT_FOUND	Failed to provide a supported localInstrument in PI Consent.	"path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "ReasonCode": "NOT_FOUND", "Description": "Method not supported by Provider's API profile" }] } }	Typically occurs when TPP does not use a value supported by their API standard.	Make sure that you select one of the specified values for localInstrument field based on the specification. Also make sure that you are filling in all the dependent fields according to the specification before re-submitting the message
"Invalid currency for selected localInstrument "	FORMAT_ERROR	Failed to provide required currency details for selected localInstrument in PI consent. Consult the API specification for field dependencies.	"path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": "Payment details are not aligned with selected localInstrument", "Details": "path[0]=/payments/instructedAmount/currency" }] } }	Typically occurs when TPP does not fill in all dependent fields used by their API standard, specifically currency field with the currency value specific to localInstrument.	Make sure that you are filling in currency field with the corresponding value to the selected localInstrument according to the specification before re-submitting the message.

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
"Conditional field RemittanceInformationUnstructured is expected for selected localinstrument "	FORMAT_ERROR	Failed to provide RemittanceInformation Unstructured field in the request to an ASPSP with Polish API standard.	"path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "FORMAT_ERROR", "Description": "Request details are not aligned with API standard", "Details": "path[0]=/payments/remittanceInformationUnstructured" }] } }	Typically occurs when TPP does not fill in dependent fields for their API standard, i.e. "RemittanceInformationUnstructured".	Ensure that a value populated in RemittanceInformation Unstructured is present and re-submit the message.

6.3.17 HTTP Status 400 - PIS– Cross-border Payment Initiation Transfer request - Specific reason codes

No additional list of reason codes is returned in the context of submitting a Cross-border Payment Initiation Transfer Request (i.e. POST /payments/cross-border-credit-transfers).

Table 19 HTTP Status 400 - PIS – Cross-border Payment Initiation Transfer requests - Specific reason codes

Error Message	Error (Reason) Code	Description	Developer Details	Sample of the error	Typical Occurrences	Next Steps
"ASPSP's signature didn't pass validation. Result code: %s"	INVALID_SIGNATURE	The signature of ASPSP that was provided in the authorization code of the request couldn't be validated.	"path[i]=<path to the element that failed the validation>;" where i = 0, 1, 2, etc. for each element that failed the validation	{ "Errors": { "Error": [{ "Source": "com.mastercard.mcob.apiserver", "ReasonCode": "INVALID_SIGNATURE", "Description": "ASPSP's signature didn't pass validation. Result code: expired", "Details": "path[0]=/requestInfo/authorization" }] } }	Typically occurs because the signature of the ASPSP in the authorization string didn't pass one or multiple validations. This can be as a result of a modified redirect URL by a PSU, an expired signature, or other reasons.	Your request didn't get sent to ASPSP since the signature of the ASPSP in authorization code failed validation. Next steps depend on the result code in error message: • "revoked" – The ASPSP revoked the signature certificate used. • "expired" - signature certificate has expired. Reinitiate payment. • "unknown" - signature certificate couldn't be identified. Reinitiate payment or contact support for more details. • "unsigned" - signature certificate is missing. Reinitiate payment or contact support for more details. • "noncompliant" - signature certificate is of an unexpected format. Contact support for more details. If you think you should be able to use the specified value, contact the Open Banking Connect API support team.

6.3.18 HTTP Status 400 - Payment Initiation Service – Cross-border Payment Status request - Specific reason codes

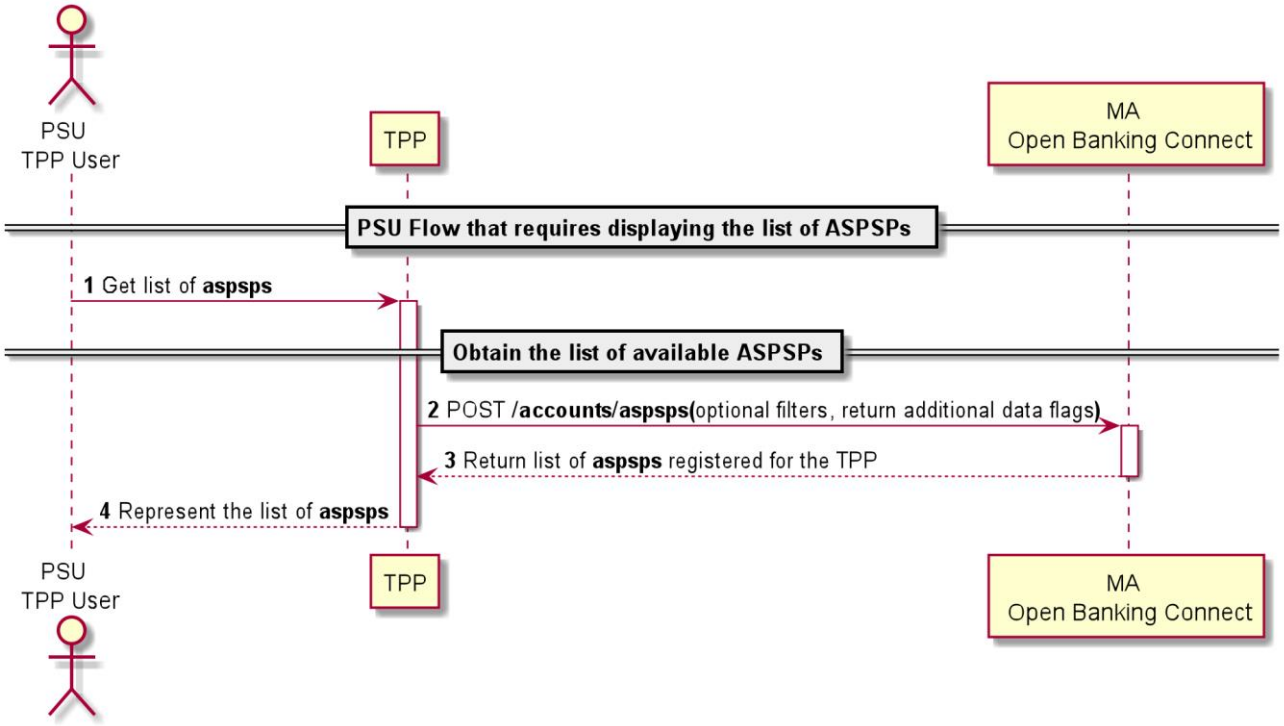
No additional list of reason codes is returned in the context of submitting a Cross-border Payment Status Request (i.e. POST /payments/cross-border-credit-transfers/payment-status). Only HTTP Status 400 – General reason codes is returned.

7 Account Information Service

7.1 Get List of ASPSPs Request

The goal of this request is to retrieve (based on the Client ID) the list of available ASPSPs.

Image 19: Account Information Service – Get List of ASPSPs Sequence Diagram



Request

```
POST /accounts/aspsps/
```

Request header

N/A

Request body

Table 20 Get List of ASPSPs Request Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
requestInfo	Includes information about request being processed	MC	This element encapsulates all request information sent to the API Service.	M	1..1	Object	Set of elements used to define the request details	-
++ xRequestId	ID of the request, unique to the call, as determined by the initiating party (TPP)	TPP	A memorable ID could be used to support in a dispute	M	1..1	UUID	Free field that allows for the addition of information that can be referenced for future use.	36
id	ID of a financial institution servicing the Accounts of the PSU	MC	A filter parameter could be used to filter ASPSPs by ASPSP id	O	0..1	String	Filter by internal id of ASPSP, exact match pattern	1-256
name	Name of a financial institution servicing the Accounts of the PSU	MC	A filter parameter could be used to filter ASPSPs by ASPSP name	O	0..1	String	Filter by name of ASPSP, partial matching, non-case sensitive.	1-256

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
country	Country in which a financial institution servicing the Accounts of the PSU resides	MC	A filter parameter could be used to filter ASPSPs by country code	O	0..1	String	Filter by country code (ISO 3166 Alpha 2), exact match pattern, case sensitive, upper case.	ISO 3166 ALPHA 2 CHAR
returnAdditionalData	A parameter which drives the additional fields included in message response	MC	This element is used to specify which additional fields could be included in response	O	0..1	Enum	Specifies additional fields for ASPSP object which could be included in response	Enum: ["capabilities", "logo"]

Response – Success

HTTP Response Code = 200, OK

Note For a full list of response codes, including those returned in failure scenarios, see section Response Codes.

Response header

N/A

Response body

Table 21 Get List of ASPSPs Request Response Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
originalRequestInfo	Returns original request information to the initiating party (TPP).	MC	This object contains original request information	M	1..1	Object	Original request information received from the initiating party	-
++ xRequestId	A memorable ID which could be used to support in a dispute	TPP	This element could be used for request-response tracking	M	1..1	UUID	Original xRequestId given by the client on request	36
aspsps	The list of financial institutions which holds PSU accounts	TPP	List of ASPSPs.	M	1..*	Object	Set of elements used to define the ASPSPs details	-
++ id	ID of a financial institution servicing the Accounts of the PSU	TPP	Identification of ASPSP	O	0..1	String	Specifies the identification code of a financial institution which holds PSU accounts	1-256
++ name	Name of a financial institution servicing the Accounts of the PSU	TPP	Name of ASPSP	O	0..1	String	Specifies the name of a financial institution which holds PSU accounts	1-256
++ aspspServices	List of ASPSP supported services	TPP	Name of provided services	O	0..1	String	ASPSP provided services	1-256 AIS - account information service, PIS - payment initiation service
++ profile	Depending of ASPSP supported profile different fields can be available in API endpoints	TPP	Name of supported profile	O	0..1	Enum	The Open Banking API standard implemented on the side of the ASPSP. ASPSP supported profiles: ["CMA9", "PolishAPI", "NextGenPSD2", "STET"]	Enum: ["CMA9", "PolishAPI", "NextGenPSD2", "STET"]

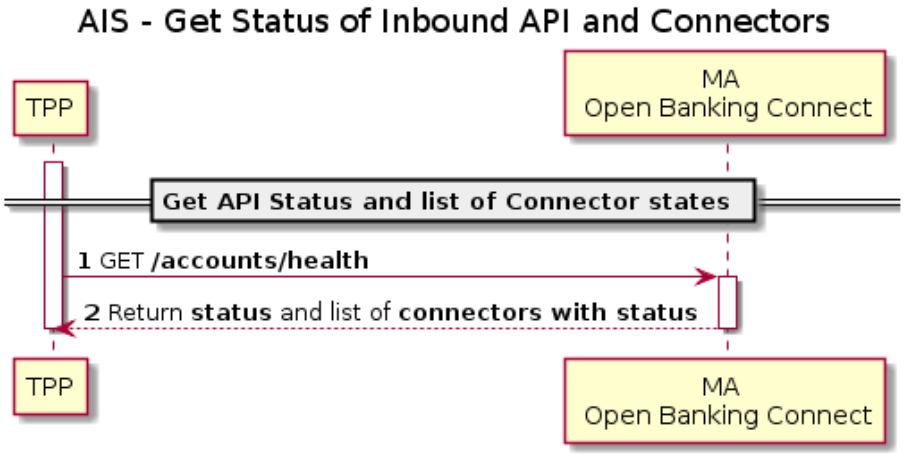
Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ country	Country in which a financial institution resides	TPP	Code of country	O	0..1	Enum	ASPSP supported country code	ISO 3166 ALPHA 2 CHAR Enum: ["GB", "PL", "AT", "BE", "BG", "DE", "DK", "EE", "ES", "FI", "FR", "IT", "GR", "HU", "HR", "NL", "LV", "LU", "RO", "SE", "SI"]
++ capabilities	Contains the list of ASPSP capabilities related to an API profile	TPP	List of capabilities	O	0..1	Object	Set of elements used to define the capabilities related to API profile assigned to ASPSP (Dictionary map)	-
++++ any<*>	Capabilities names and availability status details	TPP	Capability name and availability status	O	0..*	String	Data map <string, string> for ++ capabilities	1-256
++ logo	Contains ASPSP logo information	TPP	Logo information	O	0..1	Object	Set of elements used to define the logo metadata and image content (Dictionary map)	-
++++ fileType	File type to store logo images	TPP	Type of logo image file	O	0..1	Enum	Logo image file format. Supports the following image file formats: jpg, png, svg	Enum: [png, jpg, svg]
++++ binaryContent	Contains ASPSP's logo image in binary format	TPP	Binary content of the image	O	0..1	String	Binary content of the ASPSP logo image file in Base64 encoding (deprecated)	1-*
++++ logoURL	Returns URL link to ASPSP logo image file	TPP	URL to logo image file	O	0..1	String	The URL link to ASPSP logo image file	1-*

For more information on errors received, see the error message structure described in Error Information.

7.2 Get Status of Inbound API and AIS Connectors

The goal of this request is to check the status of Inbound API and AIS Connectors.

Image 20: Account Information Service – Get Status of Inbound API and AIS Connectors Sequence Diagram



Request

```
GET /accounts/health
```

Request header

N/A

Request body

N/A

Response – Success

HTTP Response Code = 200, OK

Note For a full list of response codes, including those returned in failure scenarios, see section Response Codes.

Response header

N/A

Response body

Table 22 Get Status of Inbound API and Connectors Request Response Body

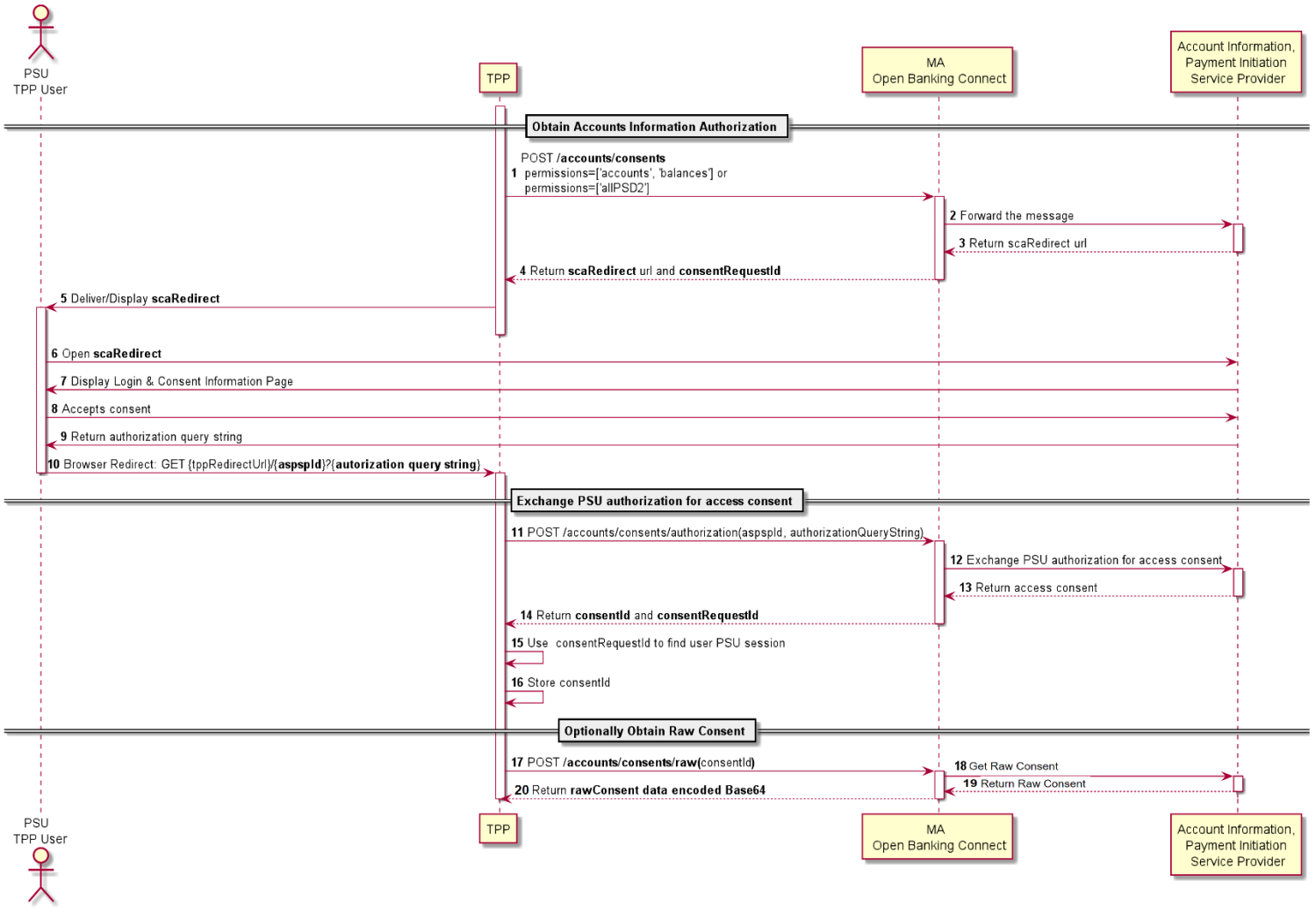
Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
status	Returns health status of inbound API	MC	Status of inbound API	M	1..1	String	Inbound API status. Status values: {UP, DOWN, DEGRADED}	1-256
statusMessage	Returns status message of inbound API	MC	Status Message of inbound API	O	0..1	String	Additional details of the status (e.g. if Down, what is down). Currently not populated.	1-256
connectors	Contains the list of OB connectors used to connect to partners and their statuses	MC	List of connectors	O	1..*	Array	Array that contains the connector of OB Connect and the status of each	-
++ name	Name of connector	MC	Name of connector	M	1..1	String	Connector name	1-256
++ status	Returns connector status	MC	Status of connector	M	1..1	String	Connector status. Status values: {UP, DOWN, DEGRADED}	1-256
++ statusMessage	Returns status message of connector	MC	Status Message of connector	O	0..1	String	Additional details of the status (e.g. if Down, what is down). Currently not populated.	1-256

For more information on errors received, see the error message structure described in Error Information.

7.3 Get Raw Consent Request

The goal of this request is to retrieve, based on the Consent ID previously obtained, the raw consent data. This flow assumes the PSU has already consented to this access. For more information, see section Account Information Consent Request.

Image 21: Account Information Service – Get Raw Consent Sequence Diagram



Request

Table 23 Get Raw Consent Request

API Endpoint	POST /accounts/consents/raw
API Profile	CMA9

Request header

N/A

Request body

Table 24 Get Raw Consent Request Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
requestInfo	Includes information about request being processed	MC	This element encapsulates all request information sent to the API Service.	M	1..1	Object	Set of elements used to define the request details	-
++ xRequestId	ID of the request, unique to the call, as determined by the initiating party (TPP)	TPP	A memorable ID could be used to support in a dispute	M	1..1	UUID	Free field that allows for the addition of information that can be referenced for future use.	36
++ consentId	ID of consent, unique to the call, as determinate by ASPSP after PSU authorization	ASPSP	Consent ID received from ASPSP after consent authorisation	M	1..1	String	Unique identification as assigned to identify the account access consent resource.	1-256
++ aspspId	ID of a financial institution servicing the Accounts of the PSU	ASPSP	Identification of ASPSP	M	1..1	UUID	Specifies the identification code of a financial institution which holds PSU accounts	36

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ merchant	Collect merchant data for reporting purposes	MC	Merchant data used for reporting or reconciliation purposes	O	0..1	Object	Set of elements used to define the merchant details	-
++++ id	Merchant identification code to identify the merchant	MC	Unique Merchant identifier per TPP, which could be used for reporting or reconciliation purposes	M	1..1	String	Field is available for TPPs to enable capturing of a merchant ID	1-256
++++ name	Merchant name to identify the merchant	MC	Name of merchant	M	1..1	String	Field is available for TPPs to enable capturing of a merchant name	1-256

Response – Success

HTTP Response Code = 200, OK

Note For a full list of response codes, including those returned in failure scenarios, see section Response Codes.

Response header

N/A

Response body

Table 25 Get Raw Consent Request Response Body

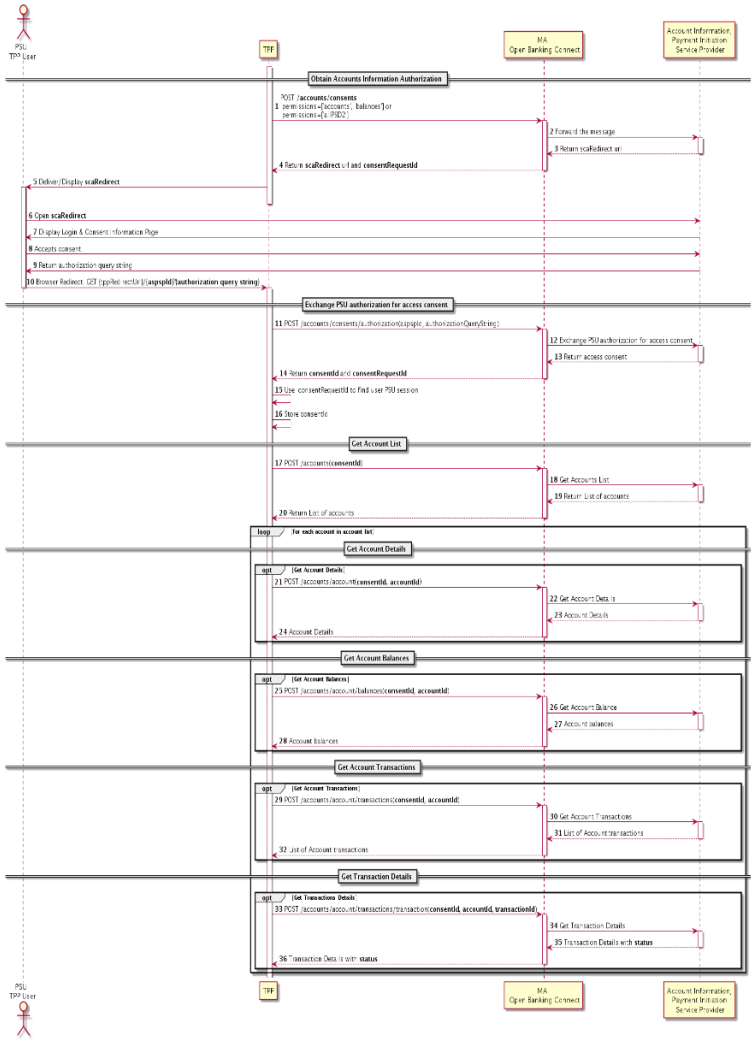
Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
originalRequestInfo	Returns original request information to the initiating party (TPP).	MC	This object contains original request information	M	1..1	Object	Original request information received from the initiating party	-
++ xRequestId	A memorable ID which could be used to support in a dispute	TPP	This element could be used for request-response tracking	M	1..1	UUID	Original xRequestId given by the client on request	36
++ rawConsent	Returns raw consent data from associated ASPSP	TPP	Raw consent data could be used to extract additional information sent by ASPSP	O	0..1	String	Raw consent data received from ASPSP and encoded Base64. It is available only for specific API profile standards.	1-*

For more information on errors received, see the error message structure described in Error Information.

7.4 Account Information Consent Request

The goal of this request is to send to the ASPSP the details of the PSU's consent for the TPP to access Account Information and then obtain a URI to redirect the PSU for Authentication and Authorisation of the Consent.

Image 22: Account Information Service – Account Information Consent Sequence Diagram



Request

Table 26 Account Information Consent Request

API Endpoint	POST /accounts/consents
API Profile	CMA9, Polish API, NextGenPSD2, STET

Request header

N/A

Request body

Table 27 Account Information Consent Request Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
requestInfo	Includes information about request being processed	MC	This element encapsulates all request information sent to the API Service	M	1..1	Object	Set of elements used to define the request details	-
++ xRequestId	ID of the request, unique to the call, as determined by the initiating party (TPP)	TPP	A memorable ID could be used to support in a dispute	M	1..1	UUID	Free field that allows for the addition of information that can be referenced for future use.	36
++ tppRedirectURI	URI of the TPP, where the transaction flow shall be redirected to after a Redirect	ASPSP	This element is used to specify the URI where should be redirected PSU after consent authorization	M	1..1	String	Call back URI where to redirect the PSU after authorisation of the consent	1-*

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ aspspId	ID of a financial institution servicing the Accounts of the PSU	ASPSP	Identification of ASPSP	M	1..1	UUID	This element is used to specify the identification code of a financial institution which holds PSU accounts	36
validUntil	Specified date the account permissions will expire	ASPSP	This is the date on which the account permission consent is to be expired.	O	0..1	ISODate	Consent validity date (future use)	ISODate, YYYY-MM-DD
++ merchant	Collect merchant data for reporting purposes	MC	Merchant data used for reporting or reconciliation purposes	O	0..1	Object	Set of elements used to define the merchant details	-
++++ id	Merchant identification code to identify the merchant	MC	Unique Merchant identifier per TPP, which could be used for reporting or reconciliation purposes	M	1..1	String	Field is available for TPPs to enable capturing of a merchant ID	1-256
++++ name	Merchant name to identify the merchant	MC	Name of merchant	M	1..1	String	Field is available for TPPs to enable capturing of a merchant name	1-256
permissions	Specifies the Open Banking account access data types.	ASPSP	This is a list of the data clusters being consented by the PSU and requested for authorisation with the ASPSP.	M	1..1	Array	Includes the list of access permissions granted to a TPP for account information, i.e. populates at least one value in the array. A mix of permission can be used, e.g. ["accounts", "balances"] or ["balances", "transactions"] but not all values together. Use ["allPSD2"] if to grant all permissions.	Array: ["accounts", "balances", "transactions", "allPSD2"]

Response – Success

HTTP Response Code = 200, OK

Note For a full list of response codes, including those returned in failure scenarios, see Response Codes

Response header

N/A

Response body

Table 28 Account Information Consent Request Response Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
originalRequestInfo	Returns original request information to the initiating party (TPP).	MC	This object contains original request information	M	1..1	Object	Original request information received from the initiating party	-
++ xRequestId	A memorable ID which could be used to support in a dispute	TPP	This element could be used for request-response tracking	M	1..1	UUID	Original xRequestId given by the client on request	36
aspspSCAApproach	Type of links admitted in this response	MC	This element is used to specify the SCA link types	O	0..1	Enum	Strong Customer Authentication (SCA) Approach	Enum: ["REDIRECT"]

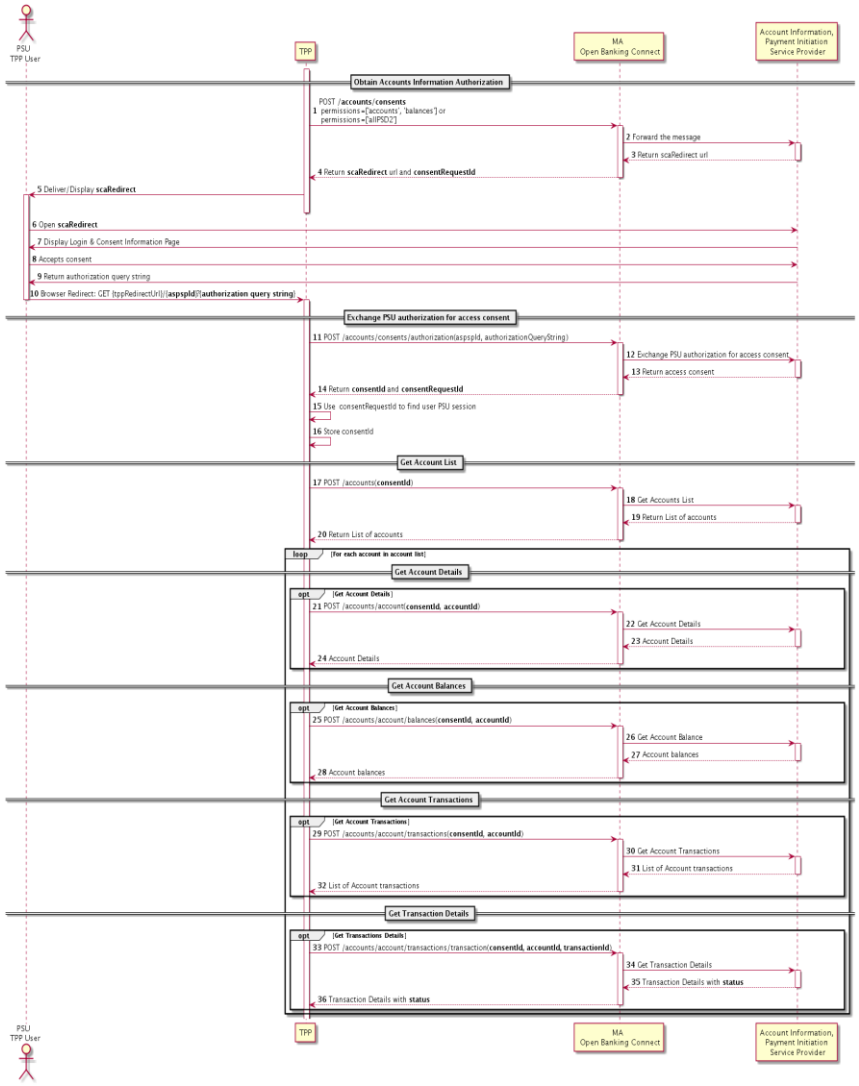
Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
consentRequestId	Unique identification as assigned by the TPP to uniquely identify the consent request.	TPP	This element could be used for consent request-response tracking	O	0..1	String	Request consent identification	1-256
_links	The list of link types admitted in response, (further links might be added for ASPSP defined extensions)	MC	The actual hyperlinks used in the response depend on the dynamical decisions of the ASPSP when processing the request.	O	0..1	Object	Set of elements used to define the links details	-
++ scaRedirect	In case of an SCA Redirect Approach, the ASPSP is transmitting the link to which to redirect the PSU browser	TPP	This element is used to specify the URL where the PSU should be redirected for consent authorization	M	1..1	URL	The URL where to redirect the PSU for authentication and authorization of the consent	1-*

For more information on errors received, see the error message structure described in Error Information.

7.5 *Exchange the PSU Authorization for Access Consent Request*

The goal of this request is to exchange the PSU authorization for access consent, which a TPP can use to access account data.

Image 23: Account Information Service – Exchange the PSU Authorization for Access Consent Sequence Diagram



Request

Table 29 Exchange the PSU authorization for access consent Request

API Endpoint	POST /accounts/consents/authorizations
API Profile	N/A

Request header

N/A

Request body

Table 30 Exchange the PSU Authorization for Access Consent Request Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
requestInfo	Includes information about request being processed	MC	This element encapsulates all request information sent to the API Service	M	1..1	Object	Set of elements used to define the request details	-
++ xRequestId	ID of the request, unique to the call, as determined by the initiating party (TPP)	TPP	A memorable ID could be used to support in a dispute	M	1..1	UUID	Free field that allows for the addition of information that can be referenced for future use.	36
++ aspspId	ID of a financial institution servicing the Accounts of the PSU	ASPSP	Identification of ASPSP	M	1..1	UUID	This element is used to specify the identification code of a financial institution which holds PSU accounts	36

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ merchant	Collect merchant data for reporting purposes	MC	Merchant data used for reporting or reconciliation purpose	O	0..1	Object	Set of elements used to define the merchant details	-
++++ id	Merchant identification code to identify the merchant	MC	Unique Merchant identifier per TPP, which could be used for reporting or reconciliation purposes	M	1..1	String	Field is available for TPPs to enable capturing of a merchant ID	1-256
++++ name	Merchant name to identify the merchant	MC	Name of merchant	M	1..1	String	Field is available for TPPs to enable capturing of a merchant name	1-256
++ authorization	Authorization data received from associated ASPSP	MC	Authorization data is used to exchange it for access consent	M	1..1	String	The authorization data received after PSU has authorized the consent	1-*

Response – Success

HTTP Response Code = 200, OK

Note For a full list of response codes, including those returned in failure scenarios, see section Response Codes.

Response header

N/A

Response body

Table 31 Exchange the PSU Authorization for Access Consent Response Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
originalRequestInfo	Returns original request information to the initiating party (TPP).	MC	This object contains original request information	M	1..1	Object	Original request information received from the initiating party	-
++ xRequestId	A memorable ID which could be used to support in a dispute	TPP	This element could be used for request-response tracking	M	1..1	UUID	Original xRequestId given by the client on request	36
++ consentRequestId	Unique identification as assigned by the TPP to uniquely identify the consent request.	TPP	This element could be used for consent request-response tracking	M	1..1	String	Request consent identification	1-256
++ consentId	ID of consent, unique to the call, as determinate by ASPSP after PSU authorization	TPP	Unique identification as assigned to identify the account access consent resource.	M	1..1	String	Access token (consent Id) received from ASPSP after consent authorisation	1-256

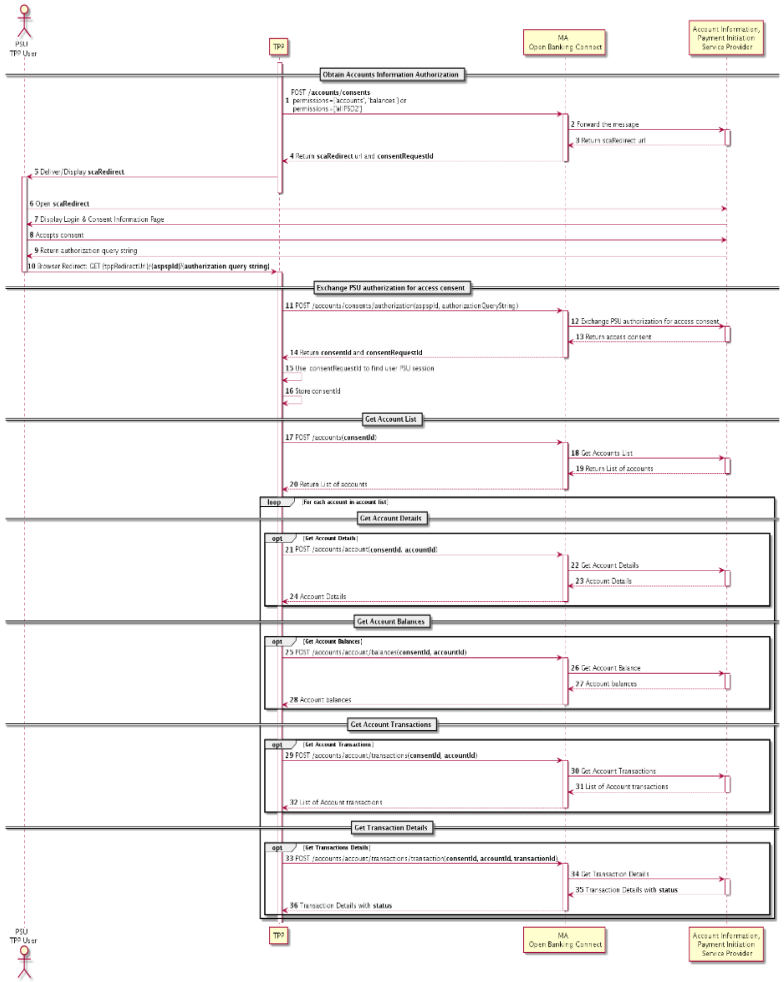
Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ signatureStatus	When TPP sends a request to a CMA9 API standard ASPSP, then MCOB will validate the signature in the authorization code. If the signature will pass validation, then the request will continue to Connectivity Partner and the response will include the successful status of signature validation, otherwise the request will be failed and TPP will receive back notification of failed signature validation	TPP	This element could be used to check TPP signature validation status	O	0..1	Enum	Status of validation of ASPSP's signature present in authorization code that TPP included in the request for credit transfer	Enum: "VALID", "UNKNOWN", "UNSIGNED", "NONCOMPLIANT"

For more information on errors received, see the error message structure described in Error Information.

7.6 *Get List of Accounts Request*

The goal of this request is to retrieve from the ASPSP, based on the Consent ID previously obtained, the list of available accounts. It is assumed the PSU has already consented to this access and the record of consent or Consent ID is stored on the ASPSP system. For more information, see section Account Information Consent Request.

Image 24: Account Information Service – Get List of Accounts Sequence Diagram



Request

Table 32 Get List of Accounts Request

API Endpoint	POST /accounts
API Profile	CMA9, Polish API, NextGenPSD2, STET

Request header

N/A

Request body

Table 33 Get List of Accounts Request Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
requestInfo	Includes information about request being processed	MC	This element encapsulates all request information sent to the API Service	M	1..1	Object	Set of elements used to define the request details	-
++ xRequestId	ID of the request, unique to the call, as determined by the initiating party (TPP)	TPP	A memorable ID could be used to support in a dispute	M	1..1	UUID	Free field that allows for the addition of information that can be referenced for future use.	36
++ consentId	ID of consent, unique to the call, as determinate by ASPSP after PSU authorization	ASPSP	Unique identification as assigned to identify the account access consent resource.	M	1..1	String	Consent ID received from ASPSP after consent authorisation	1-256

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ aspspId	ID of a financial institution servicing the Accounts of the PSU	ASPSP	Identification of ASPSP	M	1..1	UUID	This element is used to specify the identification code of a financial institution which holds PSU accounts	36
++ isLivePsuRequest	Indicate if request was actively initiated by PSU	ASPSP	Type of PSU request	O	0..1	Boolean	PSU request type	Boolean: true or false
++ psuIPAddress	The forwarded IP address field consists of the corresponding HTTP request IP address field between PSU and TPP.	ASPSP	It shall be contained if and only if the PSU actively initiated this request	C	0..1	String	IP address of PSU's terminal device.	1-256 Required when isLivePsuRequest =true
++ merchant	Collect merchant data for reporting purposes	MC	Merchant data used for reporting or reconciliation purposes	O	0..1	Object	Set of elements used to define the merchant details	-
++++ id	Merchant identification code to identify the merchant	MC	Unique Merchant identifier per TPP, which could be used for reporting or reconciliation purposes	M	1..1	String	Field is available for TPPs to enable capturing of a merchant ID	1-256
++++ name	Merchant name to identify the merchant	MC	Name of merchant	M	1..1	String	Field is available for TPPs to enable capturing of a merchant name	1-256

Response – Success

HTTP Response Code = 200, OK

Note For a full list of response codes, including those returned in failure scenarios, see Response Codes.

Response header

N/A

Response body

Table 34 Get List of Accounts Request Response Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
originalRequestInfo	Returns original request information to the initiating party (TPP).	MC	This object contains original request information	M	1..1	Object	Original request information received from the initiating party	-
++ xRequestId	A memorable ID which could be used to support in a dispute	TPP	This element could be used for request-response tracking	M	1..1	UUID	Original xRequestId given by the client on request	36
accounts	Unambiguous identification of the account to which credit and debit entries are made	TPP	Used to encapsulate accounts data	O	0..*	Object	Set of elements used to define the account details	-
++ resourceId	A unique and immutable identifier used to identify the account resource. This identifier has no meaning to the account owner.	TPP	This value should be used to make subsequent account specific requests	M	1..1	String	Account reference identification.	1-256

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ name	The account name is the name or names of the account owner(s) represented at an account level, as displayed by the ASPSP's online channels.	TPP	Account name defined by ASPSP	O	0..1	String	Account name (defined by ASPSP)	1-256
++ currency	Identification of the currency in which the account is held.	TPP	Currency of account	O	0..1	String	Account currency	ISO 4217 ALPHA 3 CHAR
++ accountHolderType	Specifies account main type: Personal or Business	TPP	Account holder type	O	0..1	Enum	Specifies the type of account (future use)	Enum: ["Business", "Personal"]
++ accountType	Specifies sub-type of the account: Savings, Credit Card, Current, etc.	TPP	Sub-type of account	O	0..1	String	Specifies the sub type of account (product family group). (future use)	1-256 Values will be returned as one of the following codes, which is aligned with ExternalCashAccountType1Code list from ISO 20022 External codes ["CACC", "CASH", "CHAR", "CISH", "COMM", "CPAC", "LLSV", "LOAN", "MGLD", "MOMA", "NREX", "ODFT", "ONDP", "OTHR", "SACC", "SLRY", "SVGS", "TAXE", "TRAN", "TRAS"]

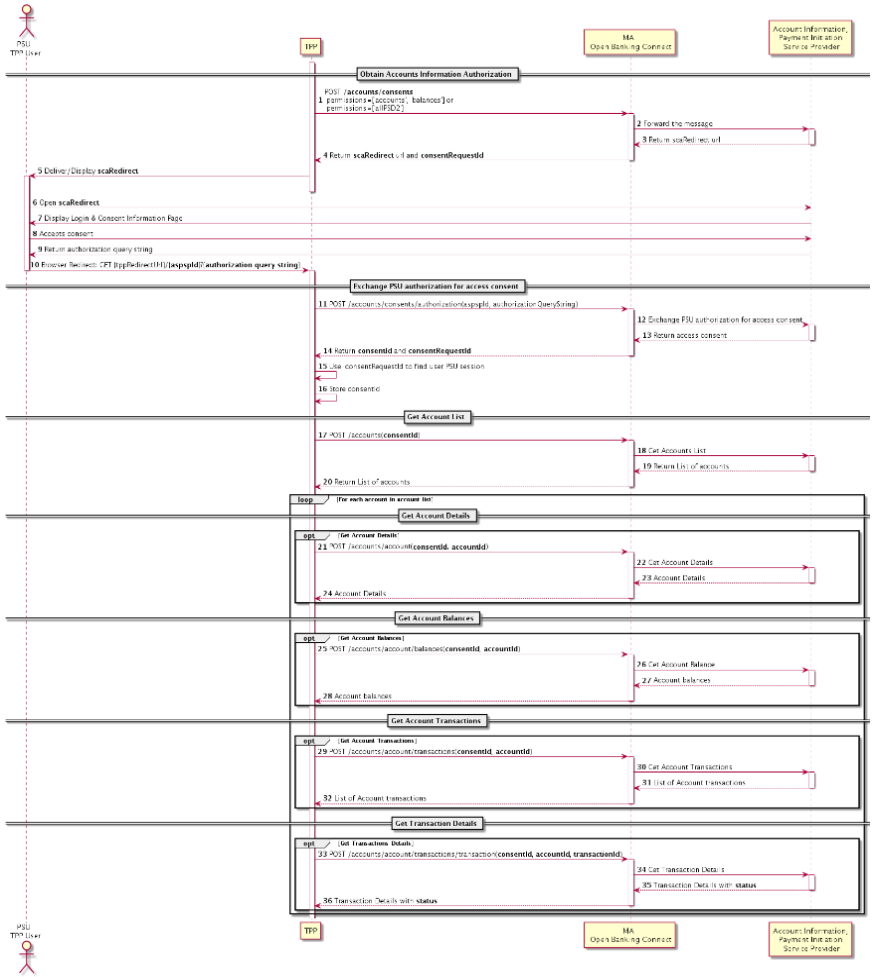
Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ nameClient	The nickname of the account, assigned by the account owner in order to provide an additional means of identification of the account.	TPP	Account name defined by account owner	O	0..1	String	Specifies the name of account defined by account owner	1-256
++ accountPsuRelations	Describes relation between PSU and account	TPP	PSU relation to account	O	0..*	Object	Description of relations between PSU and an Account	-
++++ typeOfRelation	Specifies relation between PSU and account	TPP	Can be used for filtering the results by type of relation between PSU and an Account	M	1..1	Enum	Type of relation between PSU and an Account	Enum:["Owner", "Borrower", "Guarantor", "ProxyOwner", "Beneficiary", "Trustee"]

For more information on errors received, see the error message structure described in Error Information.

7.7 Get Account Details Request

The goal of this request is to retrieve from the ASPSP, based on the Consent ID previously obtained, the account details for a specific Account ID. This flow assumes the PSU has already consented to this access and the ASPSP system stores this record of consent or Consent ID. For more information, see section Account Information Consent Request.

Image 25: Account Information Service – Get Account Details Sequence Diagram



Request

Table 35 Get Account Details Request

API Endpoint	POST /accounts/account
API Profile	CMA9, Polish API, NextGenPSD2, STET

Request header

N/A

Request body

Table 36 Get Account Details Request Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
requestInfo	Includes information about request being processed	MC	This element encapsulates all request information sent to the API Service	M	1..1	Object	Set of elements used to define the request details	-
++xRequestId	ID of the request, unique to the call, as determined by the initiating party (TPP)	TPP	A memorable ID could be used to support in a dispute	M	1..1	UUID	Free field that allows for the addition of information that can be referenced for future use.	36
++consentId	ID of consent, unique to the call, as determinate by ASPSP after PSU authorization	ASPSP	Unique identification as assigned to identify the account access consent resource.	M	1..1	String	Consent ID received from ASPSP after consent authorisation	1-256

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++aspspId	ID of a financial institution servicing the Accounts of the PSU	ASPSP	Identification of ASPSP	M	1..1	UUID	This element is used to specify the identification code of a financial institution which holds PSU accounts	36
++ isLivePsuRequest	Indicate if PSU actively initiated request	ASPSP	Type of PSU request	O	0..1	Boolean	PSU request type	Boolean: true or false
++ psuIPAddress	The forwarded IP address field consists of the corresponding HTTP request IP address field between PSU and TPP.	ASPSP	It shall be contained if and only if the PSU actively initiated this request	C	0..1	String	IP address of PSU's terminal device.	1-256 Required when isLivePsuRequest=true
++ merchant	Collect merchant data for reporting purposes	MC	Merchant data used for reporting or reconciliation purposes	O	0..1	Object	Set of elements used to define the merchant details	-
++++ id	Merchant identification code to identify the merchant	MC	Unique Merchant identifier per TPP, which could be used for reporting or reconciliation purposes	M	1..1	String	Field is available for TPPs to enable capturing of a merchant ID	1-256
++++ name	Merchant name to identify the merchant	MC	Name of merchant	M	1..1	String	Field is available for TPPs to enable capturing of a merchant name	1-256
accountId	A unique and immutable identifier used to identify the account resource.	ASPSP	Identifier of the addressed account	M	1..1	String	Account reference identification as returned in the list of accounts	1-256

Response – Success

HTTP Response Code = 200, OK

Note For a full list of response codes, including those returned in failure scenarios, see section Response Codes.

Response header

N/A

Response body

Table 37 Get Account Details Request Response Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
originalRequestInfo	Returns original request information to the initiating party (TPP).	MC	This object contains original request information	M	1..1	Object	Original request information received from the initiating party	-
++ xRequestId	A memorable ID which could be used to support in a dispute	TPP	This element could be used for request-response tracking	M	1..1	UUID	Original xRequestId given by the client on request	36
account	Unambiguous identification of the account to which credit and debit entries are made	TPP	Used to encapsulate accounts data	O	0..1	Object	Set of elements used to define the account details	-
++ resourceId	A unique and immutable identifier used to identify the account resource. This identifier has no meaning to the account owner.	TPP	This value should be used to make subsequent account specific requests	M	1..1	String	Account reference identification	1-256

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ name	The account name is the name or names of the account owner(s) represented at an account level, as displayed by the ASPSP's online channels.	TPP	Account name defined by ASPSP	O	0..1	String	Account name (defined by ASPSP)	1-256
++ currency	Identification of the currency in which the account is held.	TPP	Currency of account	O	0..1	String	Account currency	ISO 4217 ALPHA 3 CHAR
++ accountHolderType	Specifies account main type: Personal or Business	TPP	Account holder type	O	0..1	Enum	Specifies the type of account	Enum: ["Business", "Personal"]
++ accountType	Specifies sub-type of the account: Savings, Credit Card, Current, etc.	TPP	Sub-type of account	O	0..1	String	Specifies the sub type of account (product family group).	1-256 Values returned as one of the following codes, aligned with ExternalCashAccount Type1Code list from ISO 20022 External codes. ["CACC", "CASH", "CHAR ", "CISH", "COMM", "CPAC" "LLSV", "LOAN", "MGLD", "MOMA", "NREX", "ODFT", "ONDP", "OTHR", "SACC", "SLRY", "SVGS", "TAXE", "TRAN", "TRAS"]
++ nameClient	The nickname of the account, assigned by the account owner in order to provide an additional means of identification of the account.	TPP	Account name defined by account owner	O	0..1	String	Account's name client (defined by account owner)	1-256

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ holderName	Name by which a party is known, and which is usually used to identify that party.	TPP	This element is used to specify account holder name	O	0..1	String	Account Holder name (future use)	1-256
++ holderAddress	Account holder address split in separate address fields	TPP	This element encapsulates account holder address	O	0..1	Object	Set of elements used to define the holder address details (future use)	-
++++ street	Name of a street	TPP	This element is used to specify the street name	O	0..1	String	Street name	1-70
++++ buildingNumber	Number that identifies the position of a building on a street.	TPP	This element is used to specify the building number	O	0..1	String	Building Number	1-256
++++ city	Name of a built-up area, with defined boundaries, and a local government.	TPP	This element is used to specify the city name	O	0..1	String	City name	1-256
++++ postalCode	Identifier consisting of a group of letters and/or numbers that is added to a postal address to assist the sorting of mail.	TPP	This element is used to specify the post code	O	0..1	String	Postal code	1-256
++++ countrySubDivision	Identifies a subdivision of a country such as state, region, county.	TPP	This element is used to specify the county code	O	0..1	String	County name	1-256
++++ country	A nation with its own government	TPP	This element is used to specify the country code	O	0..1	String	Country name	ISO 3166 ALPHA 2 CHAR
++ holderNameAddress	Contains account holder name and address	TPP	An array which specifies account holder's name and address	O	0..4	String	Account holder's name and address	1-*

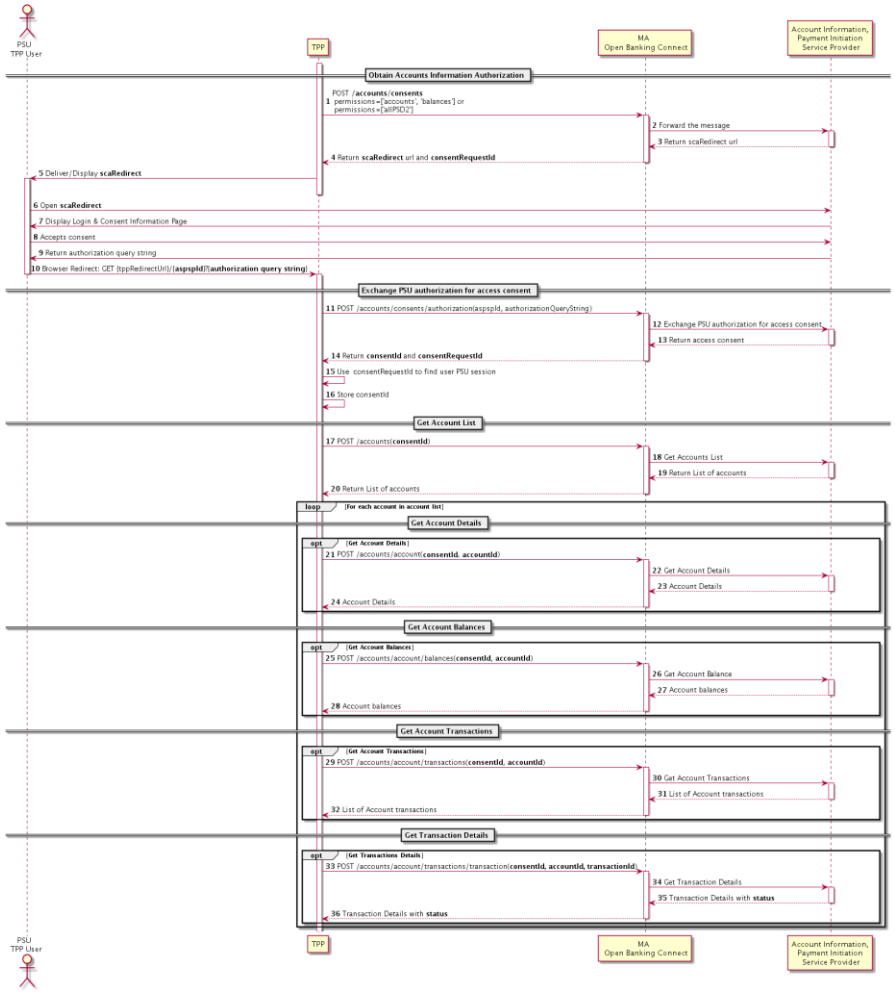
Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ accountPsuRelations	Describes relation between PSU and account	TPP	PSU relation to account	O	0..1	Object	Description of relations between PSU and an Account	-
++++ typeOfRelation	Specifies relation between PSU and account	TPP	Can be used for filtering the results by type of relation between PSU and an Account	M	1..1	Enum	Type of relation between PSU and an Account	Enum: ["Owner", "Borrower", "Guarantor", "ProxyOwner", "Beneficiary", "Trustee"]

For more information on errors received, see the error message structure described Error Information.

7.8 Get Account Balances Request

The goal of this request is to retrieve from the ASPSP, based on the Consent ID previously obtained, the account balance details for the specified Account ID. This flow assumes the PSU has already consented to this access and the ASPSP system stores this record of consent or Consent ID. For more information, see section Account Information Consent Request.

Image 26: Account Information Service – Get Account Balances Sequence Diagram



Request

Table 38 Get Account Balances Request

API Endpoint	POST /accounts/account/balances
API Profile	CMA9, Polish API, NextGenPSD2, STET

Request header

N/A

Request body

Table 39 Get Account Balances Request Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
requestInfo	Includes information about request being processed	MC	This element encapsulates all request information sent to the API Service	M	1..1	Object	Set of elements used to define the request details	-
++ xRequestId	ID of the request, unique to the call, as determined by the initiating party (TPP)	TPP	A memorable ID could be used to support in a dispute	M	1..1	UUID	Free field that allows for the addition of information that can be referenced for future use.	36
++ consentId	ID of consent, unique to the call, as determinate by ASPSP after PSU authorization	ASPSP	Unique identification as assigned to identify the account access consent resource.	M	1..1	String	Consent ID received from ASPSP after consent authorisation	1-256

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ aspspId	ID of a financial institution servicing the Accounts of the PSU	ASPSP	Identification of ASPSP	M	1..1	UUID	This element is used to specify the identification code of a financial institution which holds PSU accounts	36
++ isLivePsuRequest	Indicate if PSU actively initiated request	ASPSP	Type of PSU request	O	0..1	Boolean	PSU request type	Boolean: true or false
++ psuIPAddress	The forwarded IP address field consists of the corresponding HTTP request IP address field between PSU and TPP.	ASPSP	It shall be contained if and only if the PSU actively initiated this request	C	0..1	String	IP address of PSU's terminal device.	1-256 Required when isLivePsuRequest=true
++ merchant	Collect merchant data for reporting purposes	MC	Merchant data used for reporting or reconciliation purposes	O	0..1	Object	Set of elements used to define the merchant details	-
++++ id	Merchant identification code to identify the merchant	MC	Unique Merchant identifier per TPP, which could be used for reporting or reconciliation purposes	M	1..1	String	Field is available for TPPs to enable capturing of a merchant ID	1-256
++++ name	Merchant name to identify the merchant	MC	Name of merchant	M	1..1	String	Field is available for TPPs to enable capturing of a merchant name	1-256
accountId	A unique and immutable identifier used to identify the account resource.	ASPSP	Identifier of the addressed account	M	1..1	String	Account reference identification as returned in the list of accounts.	1-256

Response – Success

HTTP Response Code = 200, OK

Note For a full list of response codes, including those returned in failure scenarios, see section Response Codes.

Response header

N/A

Response body

Table 40 Get Account Balances Request Response Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
originalRequestInfo	Returns original request information to the initiating party (TPP).	MC	This object contains original request information	M	1..1	Object	Original request information received from the initiating party	-
++ xRequestId	A memorable ID which could be used to support in a dispute	TPP	This element could be used for request-response tracking	M	1..1	UUID	Original xRequestId given by the client on request	36
account	Unambiguous identification of the account to which credit and debit entries are made.	TPP	Used to encapsulate accounts data	O	0..1	object	Set of elements used to define the account details	-

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ resourceId	A unique and immutable identifier used to identify the account resource. This identifier has no meaning to the account owner.	TPP	This value should be used to make subsequent account specific requests	M	1..1	String	Account reference identification	1-256
++ name	The account name is the name or names of the account owner(s) represented at an account level, as displayed by the ASPSP's online channels.	TPP	Account name defined by ASPSP	O	0..1	String	Account name (defined by ASPSP)	1-256
++ balances	A list of balances regarding this account (e.g. the current balance, the last booked balance)	TPP	List of account balances	O	0..*	Object	Set of elements used to define the balances details	-
++++ balanceAmount	Amount of money of the cash balance	TPP	Balance amount details	M	1..1	Object	Set of elements used to define the balances amount details	-
++++++ currency	A code allocated to a currency by a Maintenance Agency under an international identification scheme, as described in the latest edition of the international standard ISO 4217.	TPP	The currency of the amount	M	1..1	String	Specifies the currency of the amount or of the account	ISO 4217 ALPHA 3 CHAR

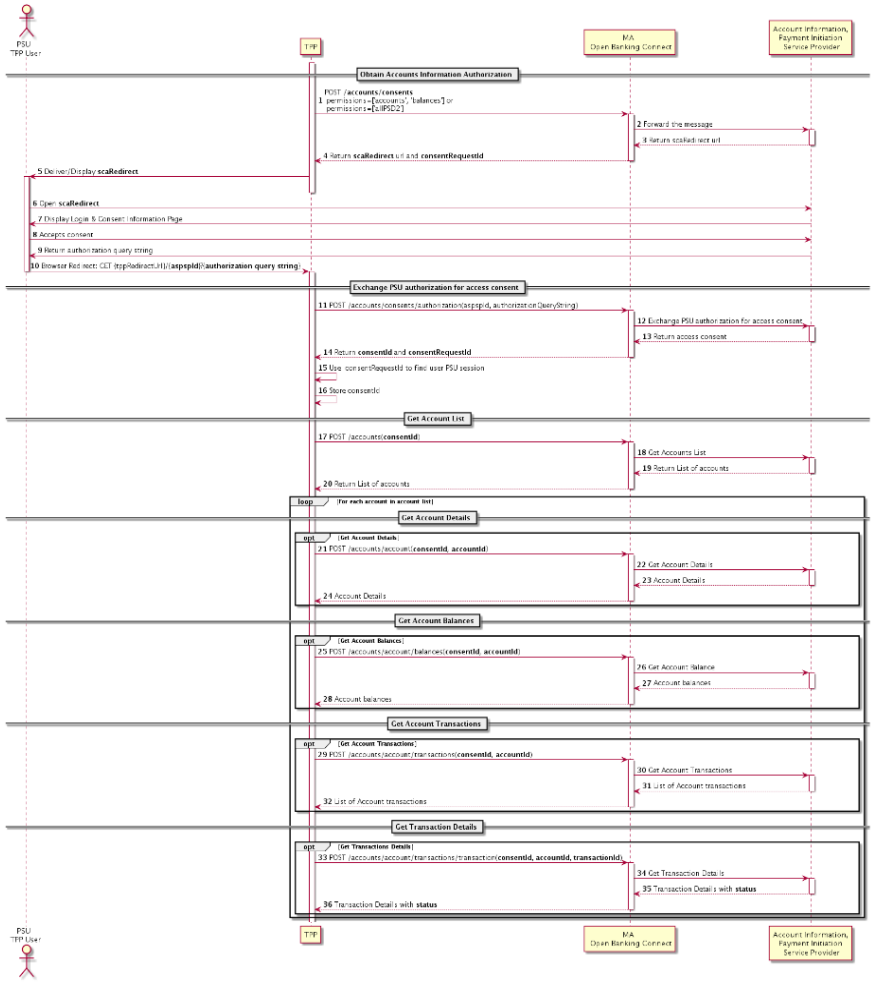
Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++++++ amount	A number of monetary units specified in an active currency where the unit of currency is explicit and compliant with ISO 4217.	TPP	Balance amount	M	1..1	Decimal	Amount of money to be moved between the debtor and creditor, before deduction of charges, expressed in the currency as ordered by the initiating party	-
++++ balanceType	Indicates balance types (these follow the standard ISO 20022 balance type enumerations)	TPP	Type of balance	O	0..1	String	Account Balance Type	1-256
++++ creditDebitIndicator	Indicates whether the balance is a credit or a debit balance	TPP	A zero balance is considered to be a credit balance.	O	0..1	Enum	Accounting flow of the amount	Enum: ["Credit","Debit"] 'Credit' if balanceAmount -> amount is greater than 0 or equal to 0 'Debit' if balanceAmount -> amount is lower than 0
++++ dateTime	Reference date for the balance	TPP	Date and time when the balance information is presented	O	0..1	ISODateTime	Indicates the date (and time) of the balance. (future use)	ISODateTime, "YYYY-MM-DDThh:mm:ssTZD"

For more information on errors received, see the error message structure described in Error Information.

7.9 Get Account Transactions Request

The goal of this request is to retrieve from the ASPSP, based on the Consent ID previously obtained, the list of transactions for the specified Account ID. This flow assumes the PSU has already consented to this access and the ASPSP system stores this record of consent or Consent ID. For more information, see section Account Information Consent Request.

Image 27: Account Information Service – Get Account Transactions Sequence Diagram



Request

Table 41 Get Account Transactions Request

API Endpoint	POST /accounts/account/transactions
API Profile	CMA9, Polish API, NextGenPSD2, STET

Request header

N/A

Request body

Table 42 Get Account Transactions Request Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
requestInfo	Includes information about request being processed	MC	This element encapsulates all request information sent to the API Service	M	1..1	Object	Set of elements used to define the request details	-
++ xRequestId	ID of the request, unique to the call, as determined by the initiating party (TPP)	TPP	A memorable ID could be used to support in a dispute	M	1..1	UUID	Free field that allows for the addition of information that can be referenced for future use.	36

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ consentId	ID of consent, unique to the call, as determinate by ASPSP after PSU authorization	ASPSP	Unique identification as assigned to identify the account access consent resource.	M	1..1	String	Consent ID received from ASPSP after consent authorisation	1-256
++ aspspId	ID of a financial institution servicing the Accounts of the PSU	ASPSP	Identification of ASPSP	M	1..1	UUID	This element is used to specify the identification code of a financial institution which holds PSU accounts	36
++ isLivePsuRequest	Indicate if request was actively initiated by PSU	ASPSP	Type of PSU request	O	0..1	Boolean	PSU request type	Boolean: true or false
++ psuIPAddress	The forwarded IP address field consists of the corresponding HTTP request IP address field between PSU and TPP.	ASPSP	It shall be contained if and only if this request was actively initiated by the PSU	C	0..1	String	IP address of PSU's terminal device.	1-256 Required when isLivePsuRequest =true
++ merchant	Collect merchant data for reporting purposes	MC	Merchant data used for reporting or reconciliation purposes	O	0..1	Object	Set of elements used to define the merchant details	-
++++ id	Merchant identification code to identify the merchant	MC	Unique Merchant identifier per TPP, which could be used for reporting or reconciliation purposes	M	1..1	String	Field is available for TPPs to enable capturing of a merchant ID	1-256
++++ name	Merchant name to identify the merchant	MC	Name of merchant	M	1..1	String	Field is available for TPPs to enable capturing of a merchant name	1-256

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
accountId	A unique and immutable identifier used to identify the account resource.	ASPSP	Identification of account resource to fetch account data	M	1..1	String	Account reference identification as returned in the list of accounts.	1-256
limit	Number of transactions to be returned in response	MC	Number of transactions on the page	M	1..1	Integer	Number of transactions returned	-
offset	Offset value from which the next set of transactions will be returned	MC	Offset value used for pagination	O	0..1	String	Offset value obtained from previous calls to get next set of transactions	1-256
bookingDateFrom	Transactions having an imputation date equal or later to this parameter are included within the result set	MC	Filter starting date (inclusive the date bookingDateFrom) used to generate the transaction list	O	0..1	Date	Initial booking date of the data range requested	Date, "YYYY-MM-DD"
bookingDateTo	Transactions having an imputation date equal or earlier to this parameter are included within the result	MC	Filter end date (inclusive the data bookingDateTo) used to generate the transaction list	O	0..1	Date	Final booking date of the data range requested	Date, "YYYY-MM-DD"

Response – Success

HTTP Response Code = 200, OK

Note For a full list of response codes, including those returned in failure scenarios, see section Response Codes.

Response header

N/A

Response body

Table 43 Get Account Transactions Request Response Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
originalRequestInfo	Returns original request information to the initiating party (TPP).	MC	This object contains original request information	M	1..1	Object	Original request information received from the initiating party	-
++ xRequestId	A memorable ID which could be used to support in a dispute	TPP	This element could be used for request-response tracking	M	1..1	UUID	Original xRequestId given by the client on request	36
offset	Offset value from which the next set of transactions should be returned	MC	Offset value used for pagination	O	0..1	String	Offset value for next call to get transactions	1-256
transactions	A list of transactions for a given account	TPP	List of transactions in account	O	0..*	Object	Set of elements used to define the transactions details	-
++ transactionId	A unique and immutable identifier used to identify the transaction resource.	TPP	Identification of transaction resource to fetch transaction data	O	0..1	String	Transaction reference identification	1-256

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ bookingDateTime	Date and time when a transaction entry is posted to an account	TPP	Booking date is the expected booking date, unless the status is booked, in which case it is the actual booking date	O	0..1	ISODateTime	Booking date and time of the transaction on the account	ISODateTime, "YYYY-MM-DDThh:mm:ssTZD"
++ transactionAmount	Amount of money of the transaction	TPP	Transaction amount details	M	1..1	Object	Set of elements used to define the transaction amount details	-
++++ currency	A code allocated to a currency by a Maintenance Agency under an international identification scheme, as described in the latest edition of the international standard ISO 4217.	TPP	The currency of the amount used in transaction	O	0..1	String	Specifies the currency of the amount	ISO 4217 ALPHA 3 CHAR
++++ amount	A number of monetary units specified in an active currency where the unit of currency is explicit and compliant with ISO 4217.	TPP	The amount of the transaction as billed to the account	M	1..1	Decimal	Amount of money to be moved between the debtor and creditor, before deduction of charges, expressed in the currency as ordered by the initiating party	-

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ remittanceInformationUnstructured	Information supplied to enable the matching/reconciliation of an entry with the items that the payment is intended to settle in an unstructured form	TPP	Description of the payment	O	0..1	String	Relevant information to the transaction in an unstructured form	1-140
++ status	Indicates a state that the transaction is currently in. Until payment is settled it progresses through different states of a transaction.	TPP	Status of the transaction	O	0..1	String	Status of a transaction entry on the books of the account servicer	1-256 Values could be returned as one of the following codes, which is aligned with ExternalPaymentTransactionStatus1Code list from ISO 20022 External codes: ["ACCC", "ACCP", "ACSC", "ACSP", "ACTC", "ACWC", "ACWP", "PDNG", "RCVD", "RJCT"]
++ creditDebitIndicator	Indicates whether the balance is a credit or a debit balance	TPP	A zero balance is considered to be a credit balance.	O	0..1	Enum	Accounting flow of the amount	Enum: ["Credit","Debit"] 'Credit' if balanceAmount -> amount is greater than 0 or equal to 0 'Debit' if balanceAmount -> amount is lower than 0

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ initiatorNameAddress	Name and Address by which an initiator party is known	TPP	This element is usually used to identify the initiator party address.	O	0..4	Array of Strings	Initiator's name and address	1-256
++ senderNameAddress	Name and Address by which a sender party is known	TPP	This element is usually used to identify the sender party address.	O	0..4	Array of Strings	Sender's name and address	1-256
++ recipientNameAddress	Name and Address by which a recipient party is known	TPP	This element is usually used to identify the recipient party address.	O	0..4	Array of Strings	Recipient's name and address	1-256
++ tradeDate	Date on which a transaction takes place or when an order is executed	TPP	Date of the operation	O	0..1	String	Date of the operation	ISODateTime, "YYYY-MM-DDThh:mm:ssTZD"
++ senderAccountNumber	Unique and unambiguous identification for the account between the account owner and the account servicer	TPP	Bank account number the money is sent from	O	0..1	String	Sender account number	1-256
++ recipientAccountNumber	Unique and unambiguous identification for the account between the account owner and the account servicer	TPP	Bank account number the money is settled	O	0..1	String	Recipient account number	1-256

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ recipientAccountMassPayment	Unique and unambiguous identification for the account between the account owner and the account servicer	TPP	Virtual bank account number the money is settled in case of mass payment operation	O	0..1	String	Recipient mass payment account number	1-256
++ recipientBankBicOrSwift	When transferring money across international lines nearly always needs to use a SWIFT/BIC code, since it's the way banks and money transfer services identify where to send money on a global scale.	TPP	The bank SWIFT/BIC code is used to identify a specific bank in an international transaction	O	0..1	String	Recipient bank BIC code	1-256
++ recipientBankName	Name to identify the financial institution	TPP	Bank name to identify the recipient's bank	O	0..1	String	Recipient bank name	1-256
++ recipientBankCode	Code to identify financial institution servicing the recipient account	TPP	Bank code to identify the recipient's bank	O	0..1	String	Recipient bank code	1-256
++ recipientBankCountryCode	Country code in which a financial institution servicing the accounts of the recipient resides	TPP	Bank country code to identify recipient's bank country residence	O	0..1	String	Recipient bank country code	1-256

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ recipientBankAddress	Information that locates and identifies a specific address, as defined by postal services, presented in free format text.	TPP	Unstructured address of the Bank to identify recipient's bank location	O	0..4	Array of Strings	Recipient bank address	1-256
++ senderAccountMassPayment	Unique and unambiguous identification for the account between the account owner and the account servicer	TPP	Virtual bank account number the money is sent from in case of mass payment operation	O	0..1	String	Sender mass payment account number	1-256
++ senderBankBicOrSwift	When transferring money across international lines nearly always needs to use a SWIFT/BIC code, since it's the way banks and money transfer services identify where to send money on a global scale.	TPP	The bank SWIFT/BIC code is used to identify a specific bank in an international transaction	O	0..1	String	Sender bank BIC/SWIFT code	1-256
++ senderBankName	Name to identify the financial institution	TPP	Bank name to identify the sender's bank	O	0..1	String	Sender bank name	1-256
++ senderBankCode	Code to identify financial institution servicing the sender account	TPP	Bank code to identify the sender's bank	O	0..1	String	Sender bank code	1-256
++ senderBankCountryCode	Country in which a financial institution servicing the accounts of the sender resides	TPP	Bank country code to identify recipient's bank country residence	O	0..1	String	Sender bank country code	1-256

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ senderBankAddress	Information that locates and identifies a specific address, as defined by postal services, presented in free format text.	TPP	Unstructured address of the Bank to identify sender's bank location.	O	0..4	Array of Strings	Sender bank address	1-256
++ transactionType	Indicates the type of transaction	TPP	The type of transaction to identify the underlying transaction	O	0..1	String	Transaction type	1-256
++ auxData	To the scope of data concerning the account and transactions, each ASPSP may add additional fields, taking advantage for this purpose of the auxData type Map field	TPP	Contains additional fields added by ASPSPS	O	0..1	Object	Set of elements used to define the additional transaction details	-
++++ any<*>	Specifies additional information about transaction reconciliation	TPP	Used by the ASPSP to transport additional transaction related information to the PSU	O	0..*	String	Data map <string, string>	1-256
++ postTransactionBalance	Specifies the account balance after the transaction is booked	TPP	Post-transaction balance	O	0..1	Decimal	Account balance after the transaction is booked	-

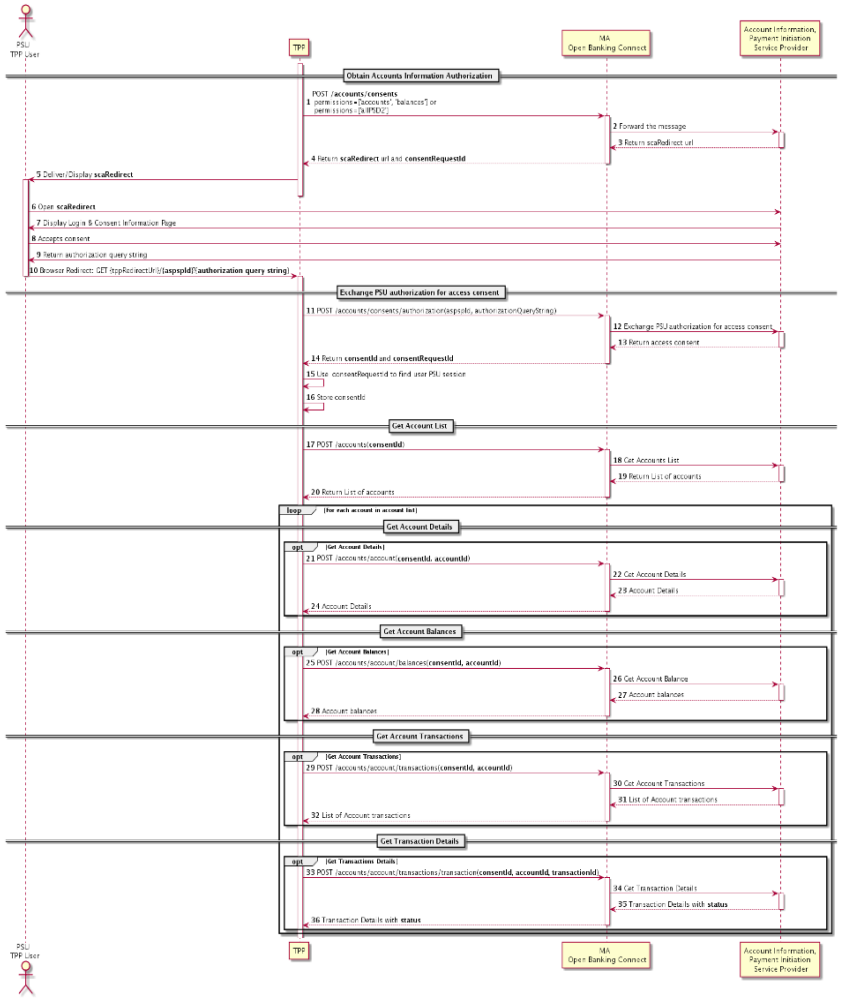
Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ mcc	Merchant Category Code of the Card Acceptor as given in the related card transaction.	TPP	Merchant Category Code for each transaction made using a card	O	0..1	String	A code of each transaction performed by card	1-256
++ rejectionReason	Specifies description of rejection reason for rejected transactions.	TPP	Used to indicate the rejection reason if a transaction has been rejected	O	0..1	String	Rejection reason	1-256
++ rejectionDate	Indicate the date when transaction has been rejected	TPP	Used to specify the rejection date if a transaction has been rejected	O	0..1	ISODateTime	Rejection date	ISODateTime, "YYYY-MM-DDThh:mm:ssTZD"
++ holdExpirationDate	Specifies transaction hold expiration date, in case if a transaction had put on hold	TPP	Hold expiration date	O	0..1	ISODateTime	Hold expiration date	ISODateTime, "YYYY-MM-DDThh:mm:ssTZD"
messages	Messages to inform TPPs in case of errors.	TPP	List of messages	O	0..1	Array	Set of elements used to define the messages details	-
++ code	Unique and unambiguous identification for the message	TPP	Code of message	O	1..*	String	Unique code of the message	1-256
++ description	Text description of returned message	TPP	Description of message	O	1..*	String	Detailed description of the message	1-256

For more information on errors received, see the error message structure described in Error Information.

7.10 Get Transaction Details Request

The goal of this request is to retrieve from the ASPSP, based on the Consent ID previously obtained, the list of transactions for the specified Account ID and Transaction ID. This flow assumes the PSU has already consented to this access and the ASPSP system stores this record of consent or Consent ID. For more information, see section Account Information Consent Request.

Image 28: Account Information Service – Get Transaction Details Sequence Diagram



Request

Table 44 Get Transaction Details Request

API Endpoint	POST /accounts/account/transactions/transaction
API Profile	Polish API, NextGenPSD2, STET

Request header

N/A

Request body

Table 45 Get Transaction Details Request Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
requestInfo	Includes information about request being processed	MC	This element encapsulates all request information sent to the API Service	M	1..1	Object	Set of elements used to define the request details	-
++ xRequestId	ID of the request, unique to the call, as determined by the initiating party (TPP)	TPP	A memorable ID could be used to support in a dispute	M	1..1	UUID	Free field that allows for the addition of information that can be referenced for future use.	36

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ consentId	ID of consent, unique to the call, as determinate by ASPSP after PSU authorization	ASPSP	Unique identification as assigned to identify the account access consent resource.	M	1..1	String	Consent ID received from ASPSP after consent authorisation	1-256
++ aspspId	ID of a financial institution servicing the Accounts of the PSU	ASPSP	Identification of ASPSP	M	1..1	UUID	This element is used to specify the identification code of a financial institution which holds PSU accounts	36
++ isLivePsuRequest	Indicate if request was actively initiated by PSU	ASPSP	Type of PSU request	O	0..1	Boolean	PSU request type	Boolean: true or false
++ psuIPAddress	The forwarded IP address field consists of the corresponding HTTP request IP address field between PSU and TPP.	ASPSP	It shall be contained if and only if this request was actively initiated by the PSU	C	0..1	String	IP address of PSU's terminal device.	1-256 Required when isLivePsuRequest= true
++ merchant	Collect merchant data for reporting purposes	MC	Merchant data used for reporting or reconciliation purposes	O	0..1	Object	Set of elements used to define the merchant details	-
++++ id	Merchant identification code to identify the merchant	MC	Unique Merchant identifier per TPP, which could be used for reporting or reconciliation purposes	M	1..1	String	Field is available for TPPs to enable capturing of a merchant ID	1-256
++++ name	Merchant name to identify the merchant	MC	Name of merchant	M	1..1	String	Field is available for TPPs to enable capturing of a merchant name	1-256

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
accountId	A unique and immutable identifier used to identify the account resource.	ASPSP	Identification of account resource to fetch account data	M	1..1	String	Account reference identification as returned in the list of accounts.	1-256
transactionId	A unique and immutable identifier used to identify the transaction resource.	ASPSP	Identification of transaction resource to fetch transaction data	M	1..1	String	Transaction reference identification	1-256

Response – Success

HTTP Response Code = 200, OK

Note For a full list of response codes, including those returned in failure scenarios, see section Response Codes.

Response header

N/A

Response body

Table 46 Get Transactions Details Request Response Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
originalRequestInfo	Returns original request information to the initiating party (TPP).	MC	This object contains original request information	M	1..1	object	Original request information received from the initiating party	-

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ xRequestId	A memorable ID which could be used to support in a dispute	TPP	This element could be used for request-response tracking	M	1..1	String	Original xRequestId given by the client on request	36
transactionId	A unique and immutable identifier used to identify the transaction resource.	ASPSP	Identification of transaction resource to fetch transaction data	O	0..1	String	Transaction reference identification	1-256
bookingDateTime	Date and time when a transaction entry is posted to an account	TPP	Booking date is the expected booking date, unless the status is booked, in which case it is the actual booking date	O	0..1	ISODateTime	Transaction datetime	ISODateTime, "YYYY-MM-DDThh:mm:ssTZD"
transactionAmount	Amount of money of the transaction	TPP	Transaction amount details	M	1..1	Object	Set of elements used to define the transaction amount details	-
++ currency	A code allocated to a currency by a Maintenance Agency under an international identification scheme, as described in the latest edition of the international standard ISO 4217.	TPP	The currency of the amount used in transaction	O	0..1	String	Specifies the currency of the amount	ISO 4217 ALPHA 3 CHAR

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ amount	A number of monetary units specified in an active currency where the unit of currency is explicit and compliant with ISO 4217.	TPP	The amount of the transaction as billed to the account	M	1..1	Decimal	Amount of money to be moved between the debtor and creditor, before deduction of charges, expressed in the currency as ordered by the initiating party	-
remittanceInformationUnstructured	Information supplied to enable the matching/reconciliation of an entry with the items that the payment is intended to settle in an unstructured form	TPP	Description of the payment	O	0..1	String	Relevant information to the transaction in an unstructured form	1-140
status	Indicates a state that the transaction is currently in. Until payment is settled it progresses through different states of a transaction.	TPP	Status of the transaction	O	0..1	String	Status of a transaction entry on the books of the account servicer	1-256 Values could be returned as one of the following codes, which is aligned with ExternalPaymentTransactionStatus1Code list from ISO 20022 External codes: ["ACCC", "ACCP", "ACSC", "ACSP", "ACTC", "ACWC", "ACWP", "PDNG", "RCVD", "RJCT"]

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
creditDebitIndicator	Indicates whether the balance is a credit or a debit balance	TPP	A zero balance is considered to be a credit balance.	M	1..1	Enum	Accounting flow of the amount	Enum: ["Credit","Debit"] 'Credit' if balanceAmount -> amount is greater than 0 or equal to 0 'Debit' if balanceAmount -> amount is lower than 0
initiatorNameAddress	Name and Address by which an initiator party is known	TPP	This element is usually used to identify the initiator party address.	O	0..4	String	Initiator's name and address	1-256
senderNameAddress	Name and Address by which a sender party is known	TPP	This element is usually used to identify the sender party address.	O	0..4	String	Sender's name and address	1-256
recipientNameAddress	Name and Address by which a recipient party is known	TPP	This element is usually used to identify the recipient party address.	O	0..4	String	Recipient's name and address	1-256
tradeDate	Date on which a transaction takes place or when an order is executed	TPP	Date of the operation	O	0..1	Date	Date of the operation	ISODateTime, "YYYY-MM-DDThh:mm:ssTZD"
senderAccountNumber	Unique and unambiguous identification for the account between the account owner and the account servicer	TPP	Bank account number the money is sent from	O	0..1	String	Sender account number	1-256

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
recipientAccountNumber	Unique and unambiguous identification for the account between the account owner and the account servicer	TPP	Bank account number the money is settled	O	0..1	String	Recipient account number	1-256
recipientAccountMassPayment	Unique and unambiguous identification for the account between the account owner and the account servicer	TPP	Virtual bank account number the money is settled in case of mass payment operation	O	0..1	String	Recipient mass payment account number	1-256
recipientBankBicOrSwift	When transferring money across international lines nearly always needs to use a SWIFT/BIC code, since it's the way banks and money transfer services identify where to send money on a global scale.	TPP	The bank SWIFT/BIC code is used to identify a specific bank in an international transaction	O	0..1	String	Recipient bank BIC/SWIFT code	1-256
recipientBankName	Name to identify the financial institution	TPP	Bank name to identify the recipient's bank	O	0..1	String	Recipient bank name	1-256
recipientBankCode	Code to identify financial institution servicing the recipient account	TPP	Bank code to identify the recipient's bank	O	0..1	String	Recipient bank code	1-256
recipientBankCountryCode	Country code in which a financial institution servicing the accounts of the recipient resides	TPP	Bank country code to identify recipient's bank country residence	O	0..1	String	Recipient bank country code	1-256

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
recipientBankAddress	Information that locates and identifies a specific address, as defined by postal services, presented in free format text.	TPP	Unstructured address of the Bank to identify recipient's bank location	O	0..4	String	Recipient bank address	1-256
senderAccountMassPayment	Unique and unambiguous identification for the account between the account owner and the account servicer	TPP	Virtual bank account number the money is sent from in case of mass payment operation	O	0..1	String	Sender mass payment account number	1-256
senderBankBicOrSwift	When transferring money across international lines nearly always needs to use a SWIFT/BIC code, since it's the way banks and money transfer services identify where to send money on a global scale.	TPP	The bank SWIFT/BIC code is used to identify a specific bank in an international transaction	O	0..1	String	Sender bank BIC/SWIFT code	1-256
senderBankName	Name to identify the financial institution	TPP	Bank name to identify the sender's bank	O	0..1	String	Sender bank name	1-256
senderBankCode	Code to identify financial institution servicing the sender account	TPP	Bank code to identify the sender's bank	O	0..1	String	Sender bank code	1-256
senderBankCountryCode	Country in which a financial institution servicing the accounts of the sender resides	TPP	Bank country code to identify recipient's bank country residence	O	0..1	String	Sender bank country code	1-256

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
senderBankAddress	Information that locates and identifies a specific address, as defined by postal services, presented in free format text.	TPP	Unstructured address of the Bank to identify sender's bank location.	O	0..4	String	Sender bank address	1-256
transactionType	Indicates the type of transaction	TPP	The type of transaction to identify the underlying transaction	O	0..1	String	Transaction type	1-256
auxData	To the scope of data concerning the account and transactions, each ASPSP may add additional fields, taking advantage for this purpose of the auxData type Map field	TPP	Contains additional fields added by ASPSPS	O	0..1	Object	Set of elements used to define the additional transaction details	-
++ any<*>	Specifies additional information about transaction reconciliation	TPP	Used by the ASPSP to transport additional transaction related information to the PSU	O	0..*	String	Data map <string, string>	1-256
postTransactionBalance	Specifies the account balance after the transaction is booked	TPP	Post-transaction balance	O	0..1	Decimal	Account balance after the transaction is booked	-

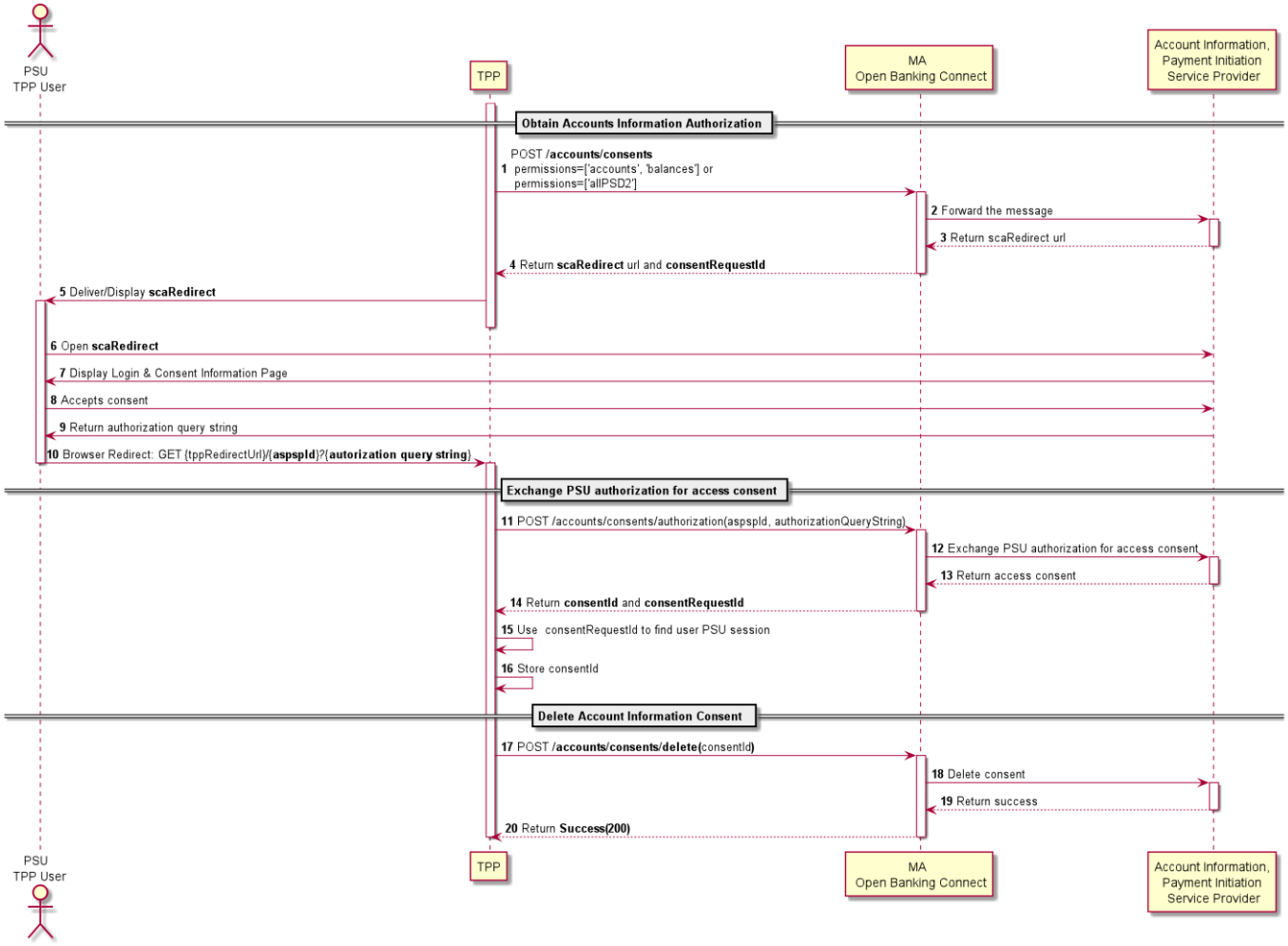
Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
mcc	Merchant Category Code of the Card Acceptor as given in the related card transaction.	TPP	Merchant Category Code for each transaction made using a card	O	0..1	String	A code of each transaction performed by card	1-256
rejectionReason	Specifies description of rejection reason for rejected transactions.	TPP	Used to indicate the rejection reason if a transaction has been rejected	O	0..1	String	Rejection reason	1-256
rejectionDate	Indicate the date when transaction has been rejected	TPP	Used to specify the rejection date if a transaction has been rejected	O	0..1	ISODateTime	Rejection date	ISODateTime, "YYYY-MM-DDThh:mm:ssTZD"
holdExpirationDate	Specifies transaction hold expiration date, in case if a transaction had put on hold	TPP	Hold expiration date	O	0..1	ISODateTime	Hold expiration date	ISODateTime, "YYYY-MM-DDThh:mm:ssTZD"

For more information on errors received, see the error message structure described in Error Information.

7.11 Delete Account Information Consent Request

The goal of this request is the deletion of consent from the ASPSP that previously created consent, so that the AI information cannot be accessed afterword. This flow assumes the PSU has already consented to this access and the ASPSP system stores this record of consent or Consent ID. For more information, see section Account Information Consent Request.

Image 29: Account Information Service – Delete Account Information Consent Diagram



Request

Table 47 Delete Account Information Consent Request

API Endpoint	POST /accounts/consents/delete
API Profile	CMA9, Polish API, NextGenPSD2, STET

Request header

N/A

Request body

Table 48 Delete Account Information Consent Request Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
requestInfo	Includes information about request being processed	MC	This element encapsulates all request information sent to the API Service	M	1..1	Object	Set of elements used to define the request details	-
++ xRequestId	ID of the request, unique to the call, as determined by the initiating party (TPP)	TPP	A memorable ID could be used to support in a dispute	M	1..1	UUID	Free field that allows for the addition of information that can be referenced for future use.	36
++ aspspId	ID of a financial institution servicing the Accounts of the PSU	ASPSP	Identification of ASPSP	M	1..1	UUID	This element is used to specify the identification code of a financial institution which holds PSU accounts	36

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ merchant	Collect merchant data for reporting purposes	MC	Merchant data used for reporting or reconciliation purposes	O	0..1	Object	Set of elements used to define the merchant details	-
++++ id	Merchant identification code to identify the merchant	MC	Unique Merchant identifier per TPP, which could be used for reporting or reconciliation purposes	M	1..1	String	Field is available for TPPs to enable capturing of a merchant ID	1-256
++++ name	Merchant name to identify the merchant	MC	Name of merchant	M	1..1	String	Field is available for TPPs to enable capturing of a merchant name	1-256
consentId	ID of consent, unique to the call, as determinate by ASPSP after PSU authorization	ASPSP	Unique identification as assigned to identify the account access consent resource.	M	1..1	String	Consent ID received from ASPSP after consent authorisation	1-256

Response – Success

HTTP Response Code = 200, OK

Note For a full list of response codes, including those returned in failure scenarios, see section Response Codes.

Response header

N/A

Response body

Table 49 Delete Account Information Consent Request Response Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
originalRequestInfo	Returns original request information to the initiating party (TPP).	MC	This object contains original request information	M	1..1	Object	Original request information received from the initiating party	-
++ xRequestId	A memorable ID which could be used to support in a dispute	TPP	This element could be used for request-response tracking	M	1..1	UUID	Original xRequestId given by the client on request	36

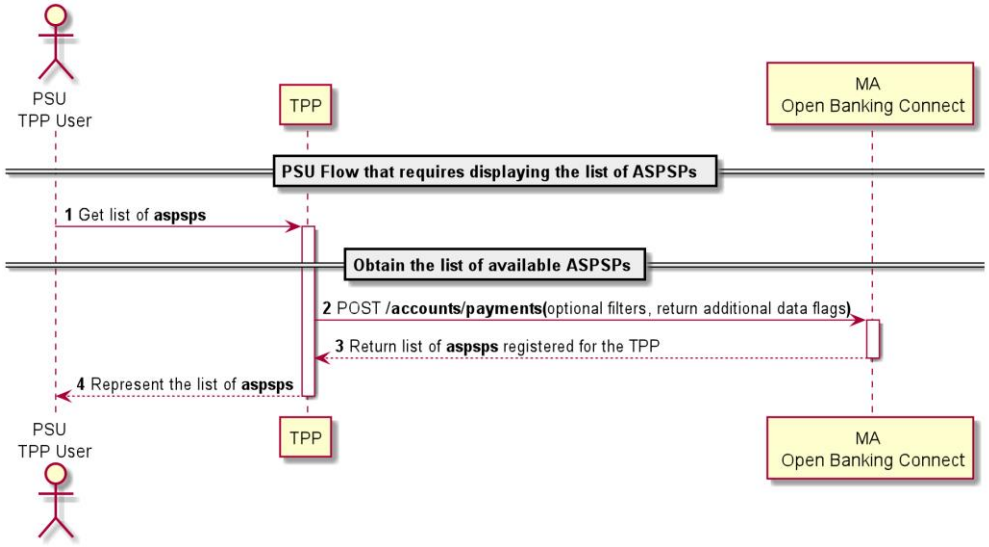
For more information on errors received, see the error message structure described in Error Information.

8 Payment Initiation Service

8.1 Get List of ASPSPs Request

The goal of this request is to retrieve (based on the Client ID) the list of available ASPSPs.

Image 30: Payment Information Service – Get List of ASPSPs Consent Diagram



Request

```
POST /payments/aspsps/
```

Request header

N/A

Request body

Table 50 Get List of ASPSPs Request Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
requestInfo	Includes information about request being processed	MC	This element encapsulates all request information sent to the API Service.	M	1..1	Object	Set of elements used to define the request details	-
++ xRequestId	ID of the request, unique to the call, as determined by the initiating party (TPP)	TPP	A memorable ID could be used to support in a dispute	M	1..1	UUID	Free field that allows for the addition of information that can be referenced for future use.	36
id	ID of a financial institution servicing the Accounts of the PSU	MC	A filter parameter could be used to filter ASPSPs by ASPSP id	O	0..1	String	Filter by internal id of ASPSP, exact match pattern	1-256
name	Name of a financial institution servicing the Accounts of the PSU	MC	A filter parameter could be used to filter ASPSPs by ASPSP name	O	0..1	String	Filter by name of ASPSP, partial matching, non-case sensitive.	1-256
country	Country in which a financial institution servicing the Accounts of the PSU resides	MC	A filter parameter could be used to filter ASPSPs by country code	O	0..1	String	Filter by country code (ISO 3166 Alpha 2), exact match pattern, case sensitive, upper case.	ISO 3166 ALPHA 2 CHAR
returnAdditionalData	A parameter which drives the additional fields included in message response	MC	This element is used to specify which additional fields could be included in response	O	0..1	Enum	Specifies additional fields for ASPSP object which could be included in response	Enum: ["capabilities", "logo"]

Response – Success

HTTP Response Code = 200, OK

Note For a full list of response codes, including those returned in failure scenarios, see section Response Codes.

Response header

N/A

Response body

Table 51 Get List of ASPSPs Request Response Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
originalRequestInfo	Returns original request information to the initiating party (TPP).	MC	This object contains original request information	M	1..1	Object	Original request information received from the initiating party	-
++ xRequestId	A memorable ID which could be used to support in a dispute	TPP	This element could be used for request-response tracking	M	1..1	UUID	Original xRequestId given by the client on request	36
aspsps	The list of financial institutions which holds PSU accounts	TPP	List of ASPSPs.	M	1..*	Object	Set of elements used to define the ASPSPs details	-
++ id	ID of a financial institution servicing the Accounts of the PSU	TPP	Identification of ASPSP	O	0..1	String	Specifies the identification code of a financial institution which holds PSU accounts	1-256
++ name	Name of a financial institution servicing the Accounts of the PSU	TPP	Name of ASPSP	O	0..1	String	Specifies the name of a financial institution which holds PSU accounts	1-256
++ aspspServices	List of ASPSP supported services	TPP	Name of provided services	O	0..1	String	ASPSP provided services	1-256 AIS - account information service, PIS - payment initiation service

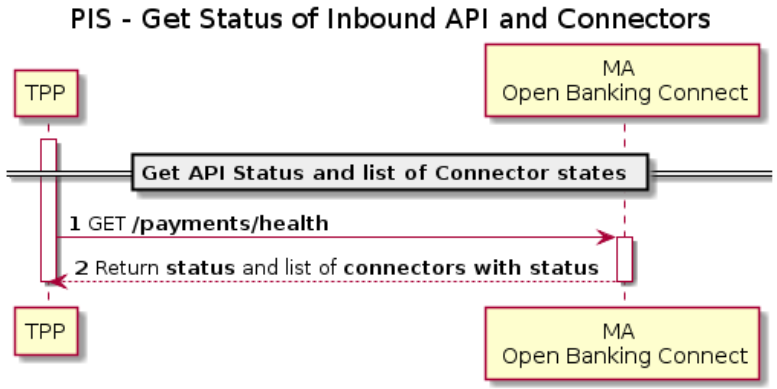
Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ profile	Depending of ASPSP supported profile different fields can be available in API endpoints	TPP	Name of supported profile	O	0..1	Enum	The Open Banking API standard implemented on the side of the ASPSP. ASPSP supported profiles: ["CMA9", "PolishAPI", "NextGenPSD2", "STET"]	Enum: ["CMA9", "PolishAPI", "NextGenPSD2", "STET"]
++ country	Country in which a financial institution resides	TPP	Country code	O	0..1	Enum	ASPSP supported country code: ["GB", "PL", "AT", "BE", "BG", "DE", "DK", "EE", "ES", "FI", "FR", "IT", "GR", "HU", "HR", "NL", "LV", "LU", "RO", "SE", "SI"]	ISO 3166 ALPHA 2 CHAR
++ capabilities	Contains the list of ASPSP capabilities related to an API profile	TPP	List of capabilities	O	0..1	Object	Set of elements used to define the capabilities related to API profile assigned to ASPSP (Dictionary map)	-
++++ any<*>	Capabilities names and availability status details	TPP	Capability name and availability status	O	0..*	String	Data map <string, string> for ++ capabilities	1-256
++ logo	Contains ASPSP logo information	TPP	Logo information	O	0..1	Object	Set of elements used to define the logo metadata and image content (Dictionary map)	-
++++ fileType	File type to store logo images	TPP	Type of logo image file	O	0..1	Enum	Logo image file format. Supports the following image file formats: jpg, png, svg	Enum: [png, jpg, svg]
++++ binaryContent	Contains ASPSP's logo image in binary format	TPP	Binary content of the image	O	0..1	String	Binary content of the ASPSP logo image file in Base64 encoding (deprecated)	1-*
++++ logoURL	Returns URL link to ASPSP logo image file	TPP	URL to logo image file	O	0..1	String	The URL link to ASPSP logo image file	1-*

For more information on errors received, see the error message structure described in Error Information.

8.2 Get Status of Inbound API and PIS Connectors

The goal of this request is to check the status of Inbound API and PIS Connectors.

Image 31: Payment Information Service – Get Status of Inbound API and PIS Connectors Diagram



Request

```
GET /payments/health
```

Request header

N/A

Request body

N/A

Response – Success

HTTP Response Code = 200, OK

Note For a full list of response codes, including those returned in failure scenarios, see section Response Codes.

Response header

N/A

Response body

Table 52 Get Status of Inbound API and Connectors Request Response Body

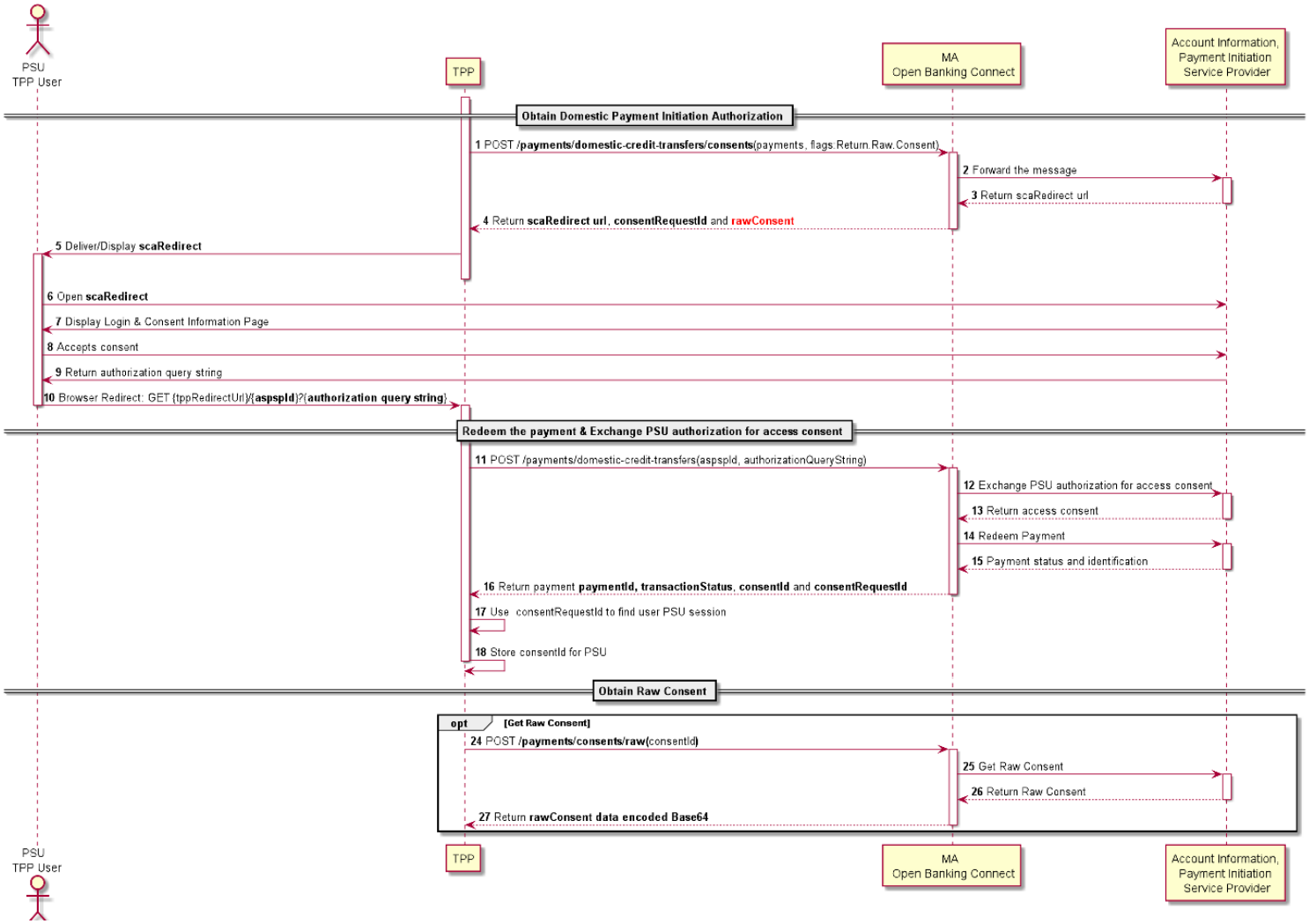
Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
status	Returns health status of inbound API	MC	Status of inbound API	M	1..1	String	Inbound API status. Status values: {UP, DOWN, DEGRADED}	1-256
statusMessage	Returns status message of inbound API	MC	Status Message of inbound API	O	0..1	String	Additional details of the status (e.g. if Down, what is down). Currently not populated.	1-256
connectors	Contains the list of OB connectors used to connect to partners and their statuses	MC	List of connectors	O	1..*	Array	Array that contains the connector of OB Connect and the status of each	-
++ name	Name of connector for partner connection	MC	Name of connector	M	1..1	String	Connector name	1-256
++ status	Returns health status of connector	MC	Status of connector	M	1..1	String	Connector status. Status values: {UP, DOWN, DEGRADED}	1-256
++ statusMessage	Returns status message of connector	MC	Status Message of connector	O	0..1	String	Additional details of the status (e.g. if Down, what is down). Currently not populated.	1-256

For more information on errors received, see the error message structure described in Error Information.

8.3 *Get Raw Consent Request*

The goal of this request is to retrieve, based on the Consent ID previously obtained, the raw consent data. This flow assumes the PSU has already consented to this access. For more information, see section Payment Initiation Consent Request.

Image 32: Payment Information Service – Get Raw Consent Diagram



Request

Table 53 Get Raw Consent Request

API Endpoint	POST /payments/consents/raw
API Profile	CMA9

Request header

N/A

Request body

Table 54 Get Raw Consent Request Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
requestInfo	Includes information about request being processed	MC	This element encapsulates all request information sent to the API Service.	M	1..1	Object	Set of elements used to define the request details	-
++ xRequestId	ID of the request, unique to the call, as determined by the initiating party (TPP)	TPP	A memorable ID could be used to support in a dispute	M	1..1	UUID	Free field that allows for the addition of information that can be referenced for future use.	36
++ consentId	ID of consent, unique to the call, as determinate by ASPSP after PSU authorization	ASPSP	Consent ID received from ASPSP after consent authorisation	M	1..1	String	Unique identification as assigned to identify the account access consent resource.	1-256

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ aspspId	ID of a financial institution servicing the Accounts of the PSU	ASPSP	Identification of ASPSP	M	1..1	UUID	Specifies the identification code of a financial institution which holds PSU accounts	36
++ merchant	Collect merchant data for reporting purposes	MC	Merchant data used for reporting or reconciliation purposes	O	0..1	Object	Set of elements used to define the merchant details	-
++++ id	Merchant identification code to identify the merchant	MC	Unique Merchant identifier per TPP, which could be used for reporting or reconciliation purposes	M	1..1	String	Field is available for TPPs to enable capturing of a merchant ID	1-256
++++ name	Merchant name to identify the merchant	MC	Name of merchant	M	1..1	String	Field is available for TPPs to enable capturing of a merchant name	1-256

Response – Success

HTTP Response Code = 200, OK

Note For a full list of response codes, including those returned in failure scenarios, see section Response Codes.

Response header

N/A

Response body

Table 55 Get Raw Consent Request Response Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
originalRequestInfo	Returns original request information to the initiating party (TPP).	MC	This object contains original request information	M	1..1	Object	Original request information received from the initiating party	-
++ xRequestId	A memorable ID which could be used to support in a dispute	TPP	This element could be used for request-response tracking	M	1..1	UUID	Original xRequestId given by the client on request	36
++ rawConsent	Returns raw consent data from associated ASPSP	TPP	Raw consent data could be used to extract additional information sent by ASPSP	O	0..1	String	Raw consent data received from ASPSP and encoded Base64.	1-*

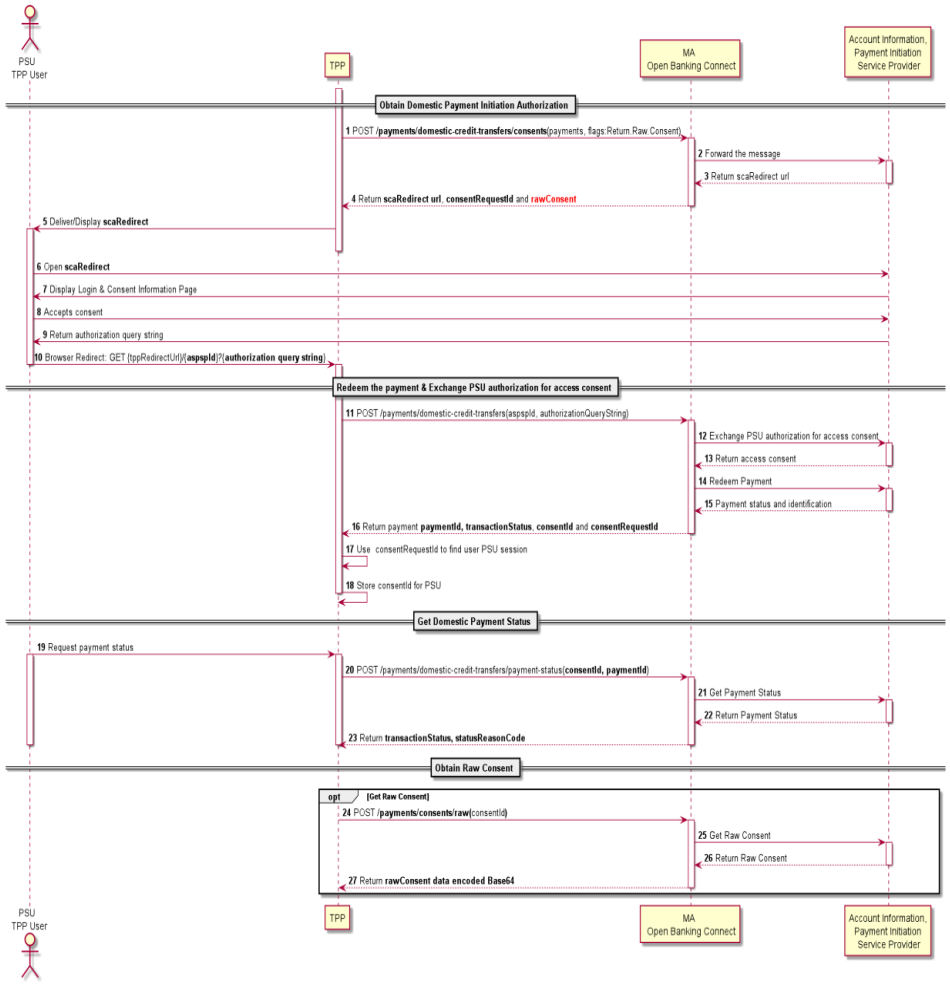
For more information on errors received, see the error message structure described in Error Information.

8.4 Domestic Payments

8.4.1 Payment Initiation Consent Request

The goal of this request is to initiate the creation of consent for the initiation of a domestic payment.

Image 33: Payment Information Service – Domestic - Payment Initiation Consent Diagram



Request

Table 56 Payment Initiation Consent Request

API Endpoint	POST /payments/domestic-credit-transfers/consents
API Profile	CMA9, Polish API

Request header

N/A

Request body

Table 57 Payment Initiation Consent Request Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
requestInfo	Includes information about request being processed	MC	This element encapsulates all request information sent to the API Service	M	1..1	Object	Set of elements used to define the request details	-
++ xRequestId	ID of the request, unique to the call, as determined by the initiating party (TPP)	TPP	A memorable ID could be used to support in a dispute	M	1..1	UUID	Free field that allows for the addition of information that can be referenced for future use.	36
++ tppRedirectURI	URI of the TPP, where the transaction flow shall be redirected to after a Redirect	ASPSP	This element is used to specify the URI where should be redirected PSU after consent authorization	M	1..1	String	Call back URI where to redirect the PSU after authorisation of the consent	1-*

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ aspspId	ID of a financial institution servicing the Accounts of the PSU	ASPSP	Identification of ASPSP	M	1..1	UUID	This element is used to specify the identification code of a financial institution which holds PSU accounts	36
++ merchant	Collect merchant data for reporting purposes	MC	Merchant data used for reporting or reconciliation purposes	O	0..1	Object	Set of elements used to define the merchant details	-
++++ id	Merchant identification code to identify the merchant	MC	Unique Merchant identifier per TPP, which could be used for reporting or reconciliation purposes	M	1..1	String	Field is available for TPPs to enable capturing of a merchant ID	1-256
++++ name	Merchant name to identify the merchant	MC	Name of merchant	M	1..1	String	Field is available for TPPs to enable capturing of a merchant name	1-256
++ flags	The list of flags which can set the additional parameters in returned data	MC	This element is used to specify additional parameters which influence data returned in response	O	0..1	Enum	Request information flags which can influence the behaviour or returned data	Enum: ['Return.Raw.Consent']
payments	The object contains the list of fields used to send a payment instruction to financial institution	MC	This element encapsulates all payment fields	M	0..1	Object	Set of elements used to define the payments details	-

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ endToEndIdentification	Unique identification assigned by the initiating party to unambiguously identify the transaction. This identification is passed on, unchanged, throughout the entire end-to-end chain.	ASPSP	The end-to-end identification can be used for reconciliation or to link tasks relating to the transaction. It can be included in several messages related to the transaction.	M	1..1	String	Field is available for TPPs to enable end-to-end identification of the transaction.	1-18
++ localInstrument	User community specific instrument	ASPSP	This element is used to specify a local instrument, local clearing option and/or further qualify the service or service level.	M	1..1	Enum	User community specific instrument specifies which clearing system should be used to process the payment instruction	Enum: ["UK.FasterPayments","PL.Elixir"]
++ instructionPriority	Indicator of the urgency or order of importance that the instructing party would like the instructed party to apply to the processing of the instruction	ASPSP	This element is used to specify urgency mode of the payment.	C	1..1	Enum	The way the transfer should be settled	Enum: ['Normal', 'Urgent'] Mandatory if localInstrument = PL.Elixir

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ debtorAccount	Unambiguous identification of the account of the debtor to which a debit entry will be made as a result of the transaction	ASPSP	Specify account in a financial institution servicing the debtor	C	0..1	Object	Set of elements used to define the debtor account details	If localInstrument = 'UK.FasterPayments' and debtorAgent was specified then becomes mandatory If localInstrument = 'PL.Elixir' and debtorAgent was specified then becomes mandatory
++++ identification	Identification assigned by an institution to identify an account. This identification is known by the account owner.	ASPSP	This element is used to specify the account identification number	M	1..1	String	Account identification which can be used on payload-level to address specific accounts	1-256 If localInstrument = 'UK.FasterPayments' and debtorAgent was specified, then this needs to be a UK Account number (8 digits)
++ debtorAgent	Unambiguous identification of the financial institution of the debtor to which a debit entry will be made as a result of the transaction	ASPSP	Specify financial institution servicing an account for the debtor	C	0..1	Object	Set of elements used to define the debtor agent details	If localInstrument = 'UK.FasterPayments' and debtorAccount was specified then becomes mandatory
++++ memberIdentification	Code allocated to a financial institution by the ISO 9362 Registration Authority to identify the financial institution	ASPSP	This element is used to specify the financial institution identification code	M	1..1	String	Identification code to identify the financial institution.	1-256 If localInstrument = 'UK.FasterPayments' and debtorAccount was specified, then this needs to be a UK Sort code (6 digits)

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ instructedAmount	Amount of money to be moved between the debtor and creditor, before deduction of charges, expressed in the currency as ordered by the initiating party.	ASPSP	This amount has to be transported unchanged through the transaction chain	M	1..1	Object	Structure aiming to embed the amount and the currency to be used	-
++++ currency	A code allocated to a currency by a Maintenance Agency under an international identification scheme, as described in the latest edition of the international standard ISO 4217 "Codes for the representation of currencies and funds"	ASPSP	The currency of the amount used in transaction	M	1..1	Enum	Specifies the currency of the amount or of the account.	Enum: ["GBP", "PLN"] GBP (mandatory if localInstrument = UK.FasterPayments) PLN (mandatory if localInstrument = PL.Elixir)
++++ amount	A number of monetary units specified in an active currency where the unit of currency is explicit and compliant with ISO 4217.	ASPSP	This element is used to specify the amount of money to be processed	M	1..1	Decimal	Amount of money to be moved between the debtor and creditor, before deduction of charges, expressed in the currency as ordered by the initiating party	Double, min: 0

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ creditorAccount	Unambiguous identification of the account of the creditor to which a credit entry will be posted as a result of the payment transaction.	ASPSP	Specify account in a financial institution servicing the creditor	M	1..1	Object	Set of elements used to define the creditor account details	-
++++ schemeName	Name of the identification scheme, in a coded form as published in an external list.	ASPSP	This field is used to identify the type of Identification used to identify an account.	C	1..1	Enum	Account scheme name	Enum ["UK.AccountNumber", "PL.AccountNumber"] UK.AccountNumber - if localInstrument = UK.FasterPayments PL.AccountNumber – if localInstrument = PL.Elixir
++++ identification	Identification assigned by an institution to identify an account. This identification is known by the account owner.	ASPSP	This element is used to specify the account identification number	M	1..1	String	Account identification which can be used on payload-level to address specific accounts	1-256

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ creditorAgent	Unambiguous identification of the financial institution of the creditor to which a credit entry will be posted as a result of the payment transaction.	ASPSP	Specify financial institution servicing an account for the creditor.	C	0..1	Object	Set of elements used to define the creditor agent details	Mandatory for 'UK.FasterPayments' and NOT used for 'PL.Elixir'
++++ clearingSystemIdentification	Name of the identification scheme, in a coded form as published in an external list.	ASPSP	This field is used to identify the type of identification used to identify an agent.	M	1..1	Enum	Clearing system identification scheme	Enum: "UK.SortCode" Mandatory if localInstrument = UK.FasterPayments
++++ memberIdentification	Code allocated to a financial institution by the ISO 9362 Registration Authority to identify the financial institution	ASPSP	This element is used to specify the financial institution identification code	M	1..1	String	Identification code to identify the financial institution.	1-256 Mandatory if localInstrument = UK.FasterPayments
++ creditorName	Name by which a party is known, and which is usually used to identify that party.	ASPSP	This element is used to specify creditor name	M	1..1	String	Name of the Beneficiary/Creditor	1-70
++ creditorAddress	Information that locates and identifies a specific address, as defined by postal services.	ASPSP	This element is usually used to identify the creditor party address.	C	1..1	Object	Set of elements used to define the creditor address details	-
++++ street	Name of a street or thoroughfare	ASPSP	This element is used to specify the street name	M	1..1	String	Street name	1-70

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++++ buildingNumber	Number that identifies the position of a building on a street.	ASPSP	This element is used to specify the building number	M	1..1	String	Building number	1-256
++++ city	Name of a built-up area, with defined boundaries, and a local government.	ASPSP	This element is used to specify the city name	M	1..1	String	City name	1-256
++++ postalCode	Identifier consisting of a group of letters and/or numbers that is added to a postal address to assist the sorting of mail.	ASPSP	This element is used to specify the post code	M	1..1	String	Postal code	1-256
++++ countrySubDivision	Identifies a subdivision of a country such as state, region, county.	ASPSP	This element is used to specify the county code	O	0..1	String	County name	1-256
++++ country	Country in which a person resides	ASPSP	This element is used to specify the country code	M	1..1	String	Country name	ISO 3166 ALPHA 2 CHAR
++ remittanceInformationUnstructured	Information supplied to enable the matching/reconciliation of an entry with the items that the payment is intended to settle, such as commercial invoices in an accounts' receivable system, in an unstructured form.	ASPSP	Description of the payment	C	0..1	String	If available, the initiating party should provide this information in the unstructured remittance form	1-140 Mandatory if localInstrument = PL.Elixir

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ requestedExecutionDate	Date at which the initiating party requests the clearing agent to process the payment.	ASPSP	This is the date on which the debtor's account is to be debited.	C	0..1	ISODate	Date when the transfer is to be executed (It is used for Future Dated payments)	ISODate, YYYY-MM-DD The requestedExecution Date field and schedule object are both optional and mutually exclusive fields
++ remittanceInformationReference	Reference information provided by the creditor to allow the identification of the underlying documents	ASPSP	If available, the initiating party should provide this reference in the structured remittance information, to enable reconciliation by the creditor upon receipt of the amount of money.	O	0..1	String	Reference information, as assigned by the creditor, to unambiguously refer to the payment transaction.	1-18
++ instructionIdentification	Unique identification as assigned by an instructing party for an instructed party to unambiguously identify the instruction.	ASPSP	The instruction identification is a point to point reference that can be used between the instructing party and the instructed party to refer to the individual instruction. It can be included in several messages related to the instruction.	O	0..1	String	Unique identification shared between the PISP and the ASPSP	1-256

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ schedule	Schedule object is used for standing orders.	ASPSP	Can be used to initiate a domestic standing orders payment instruction	C	0..1	Object	Set of elements used to define the schedule details	The schedule object and requestedExecution Date field are both optional and mutually exclusive fields
++++ startDate	The first date of execution for a standing order.	ASPSP	This element is used to specify the first payment date	M	1..1	ISODate	Payment first execution date	ISODate, YYYY-MM-DD
++++ frequency	Frequency rule for standing orders.	ASPSP	This element is used to specify the frequency of a payment	M	1..1	Enum	Payment frequency -determines how often a recurring payment will be made.	Enum: • DAIL - daily • WEEK - weekly • MNTH - monthly • QUTR - quarterly • SEMI - semi-annually • YEAR - yearly
++++ endDate	The last applicable day of execution for a given standing order. If not given, the standing order is considered as endless, until cancelled by PSU.	ASPSP	This element is used to specify payment last execution date	O	0..1	ISODate	Payment last execution date	ISODate, YYYY-MM-DD

Response – Success

HTTP Response Code = 200, OK

Note For a full list of response codes, including those returned in failure scenarios, see section Response Codes.

Response header

N/A

Response body

Table 58 Payment Initiation Consent Request Response Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
originalRequestInfo	Returns original request information to the initiating party (TPP).	MC	This object contains original request information	M	1..1	object	Original request information received from the initiating party	-
++ xRequestId	A memorable ID which could be used to support in a dispute	TPP	This element could be used for request-response tracking	M	1..1	UUID	Original xRequestId given by the client on request	36
aspspSCAApproach	Type of links admitted in this response	MC	This element is used to specify the SCA link types	O	0..1	Enum	Strong Customer Authentication (SCA) Approach	Enum: ["REDIRECT"]
consentRequestId	Unique identification as assigned by the TPP to uniquely identify the consent request.	TPP	This element could be used for consent request-response tracking	O	0..1	String	Request consent identification	1-256

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
_links	The list of link types admitted in response, (further links might be added for ASPSP defined extensions)	MC	The actual hyperlinks used in the response depend on the dynamical decisions of the ASPSP when processing the request.	O	0..1	object	Set of elements used to define the links details	-
++ scaRedirect	In case of an SCA Redirect Approach, the ASPSP is transmitting the link to which to redirect the PSU browser	TPP	This element is used to specify the URL where the PSU should be redirected for consent authorization	M	1..1	URL	The URL where to redirect the PSU for authentication and authorization of the consent	1-*
additionalData	Specifies additional parameters in returned data	TPP	This element is used to specify additional parameters returned in response	C	0..1	Object	Set of elements used to define the additional data details	-
++ rawConsent	Returns raw consent data from associated ASPSP	TPP	Raw consent data could be used to extract additional information sent by ASPSP	M	1..1	String	Raw consent data received from ASPSP and encoded Base64.	1-* Returned only if requestInfo.flags contained 'Return.Raw.Consent' value

For more information on errors received, see the error message structure described in Error Information.

After the response is received by a TPP, it will have to redirect the PSU to the URI received in 'scaRedirect' to complete the authentication with the ASPSP and authorisation of the consent.

After completing this process, the TPP will obtain an access token on the URI address specified in the 'tppRedirectURI' field in the request. Depending of the ASPSP profile, the TPP can also get additional parameters along with access token. The following table shows the list of parameters which a TPP may get from an ASPSP.

Table 59 ASPSPs parameters list

Name	Type	API Profile	Description
code	String	All Profiles	Authorization string
state	String	All Profiles	State data

Note The parameters sent from ASPSP to TPP are Base64 encoded.

The OBIE, in alignment with the security community, recommends the use of the Hybrid flow. The platform today enforces this by setting the initial response type is set to "code id_token" in the authorization request to the ASPSP's Authorization Server. The Authorization server is required to return both the access code and id_token in this case, but the manner in which it does this is left to the discretion of the Authorization Server. This could be in either form

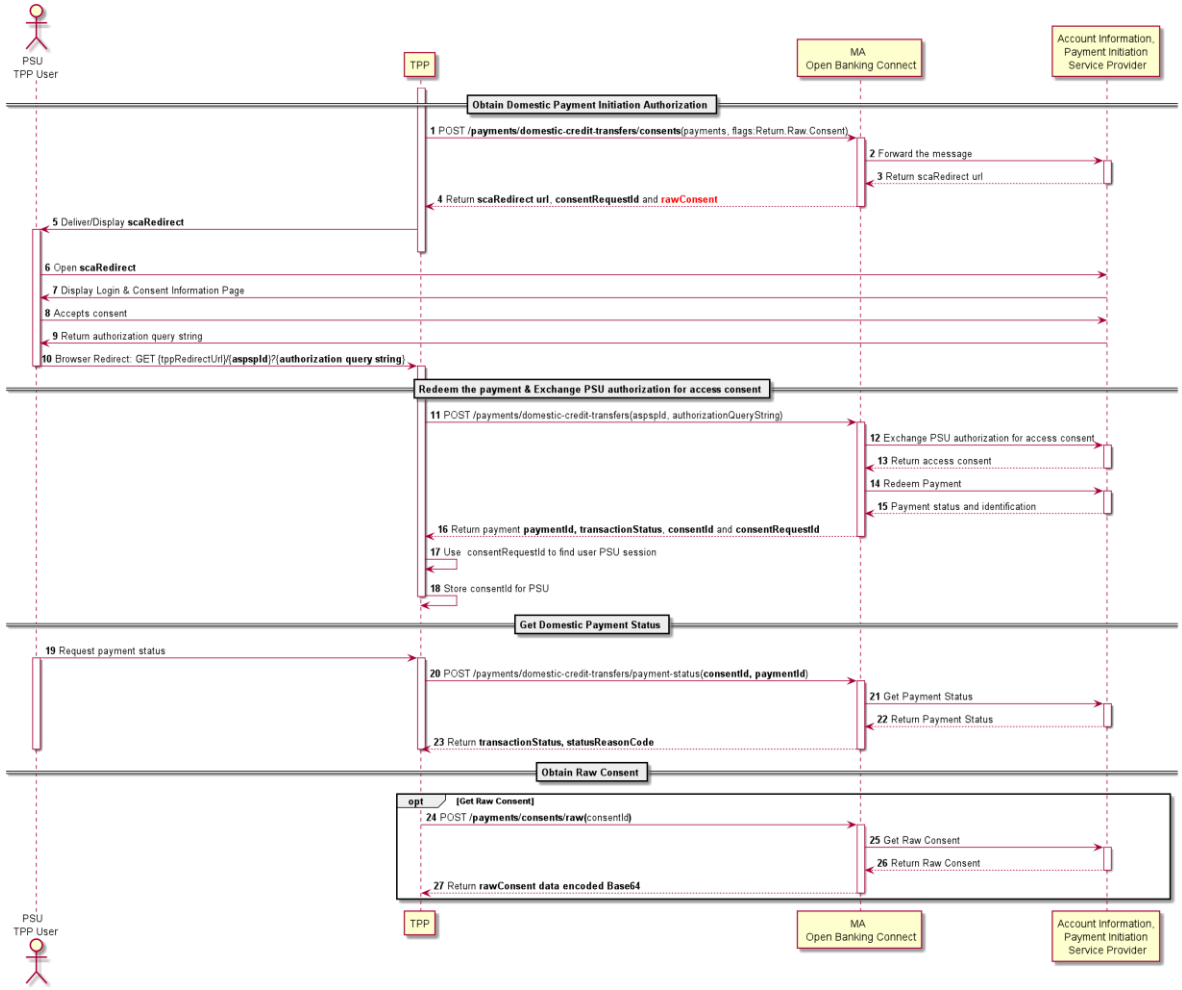
- TPP_REDIRECT_URL?All_Params
- TPP_REDIRECT_URL#All_Params

The current best practice is to return these values as a hash fragment. This allows the User Agent to receive the values and skip redirection (performance enhancement), and not send the access code via a query string parameter where it will be logged by proxies, firewalls, application servers, and other services in the application path. Having the access code logged in plain text would weaken its value. For this reason, it is suggested the application parse the access code from the hash fragment and POST it to the Client Service (TPP service) in an AJAX request, where the value may be safely ensconced within the POST body.

8.4.2 Payment Credit Transfer Initiation Request

The goal of this request is to initiate a Domestic payment on the ASPSP side using a consent that is authorised by the PSU. This flow assumes the PSU has already consented to this access and the ASPSP system stores this record of consent or Consent ID. For more information, see section Payment Initiation Consent Request.

Image 34: Payment Information Service – Domestic - Payment Credit Transfer Initiation Diagram



Request

Table 60 Payment Credit Transfer Initiation Request

API Endpoint	POST /payments/domestic-credit-transfers
API Profile	CMA9, Polish API

Request header

N/A

Request body

Table 61 Payment Credit Transfer Initiation Request Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
requestInfo	Includes information about request being processed	MC	This element encapsulates all request information sent to the API Service.	M	1..1	Object	Set of elements used to define the request details	-
++ xRequestId	ID of the request, unique to the call, as determined by the initiating party (TPP)	TPP	A memorable ID could be used to support in a dispute	M	1..1	UUID	Free field that allows for the addition of information that can be referenced for future use.	36
++ aspspId	ID of a financial institution servicing the Accounts of the PSU	ASPSP	Identification of ASPSP	M	1..1	UUID	Specifies the identification code of a financial institution which holds PSU accounts	36

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ merchant	Collect merchant data for reporting purposes	MC	Merchant data used for reporting or reconciliation purposes	O	0..1	Object	Set of elements used to define the merchant details	-
++++ id	Merchant identification code to identify the merchant	MC	Unique Merchant identifier per TPP, which could be used for reporting or reconciliation purposes	M	1..1	String	Field is available for TPPs to enable capturing of a merchant ID	1-256
++++ name	Merchant name to identify the merchant	MC	Name of merchant	M	1..1	String	Field is available for TPPs to enable capturing of a merchant name	1-256
++ authorization	Authorization data received from associated ASPSP	MC	Authorization data is used to exchange it for access consent	M	1..1	String	The authorization data received after PSU has authorized the consent	1-*

Response – Success

HTTP Response Code = 200, OK

Note For a full list of response codes, including those returned in failure scenarios, see section Response Codes.

Response header

N/A

Response body

Table 62 Payment Credit Transfer Initiation Request Response Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
originalRequestInfo	Returns original request information to the initiating party (TPP).	MC	This object contains original request information	M	1..1	object	Original request information received from the initiating party	-
++ xRequestId	A memorable ID which could be used to support in a dispute	TPP	This element could be used for request-response tracking	M	1..1	UUID	Original xRequestId given by the client on request	36
transfer	Specifies transfer information in returned data	TPP	This element encapsulates all transfer information	M	1..1	Object	Set of elements used to reference a payment instruction	-
++ paymentId	Unique identification as assigned by the ASPSP to uniquely identify the payment resource.	TPP	This value should be used to make subsequent payment specific requests	M	1..1	String	Payment identification	1-256
++ transactionStatus	Specifies the status of the payment information group	TPP	Transaction Status of the payment resource	M	1..1	Enum	Status of the payment	Enum: • ACSC • ACSP • PDNG • RCVD • RJCT • CANC

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ statusReasonCode	Provides detailed information on the status reason.	TPP	Can only be used in case the status is equal to "RJCT"	O	0..1	String	Status Reason Code is used to add more information when a generic status code is returned by transaction Status field.	The codes for statusReasonCode are defined in ISO20022 - "ExternalStatusReason1Code". The following codes could be returned when RJCT status is raised: • AM04 • CURR • AM12 • DS24 • DS02 • AG08
consent	Specifies consent information in returned data	TPP	This element encapsulates all consent information	O	0..1	Object	Set of elements used to define the consent details	-
++ consentId	ID of consent, unique to the call, as determinate by ASPSP after PSU authorization	TPP	Unique identification as assigned to identify the account access consent resource.	O	0..1	String	Access token (consent Id) received from ASPSP after consent authorisation	1-256
++ consentRequestId	Unique identification as assigned by the TPP to uniquely identify the consent request.	TPP	This element could be used for consent request- response tracking	O	0..1	String	Request consent identification	1-256

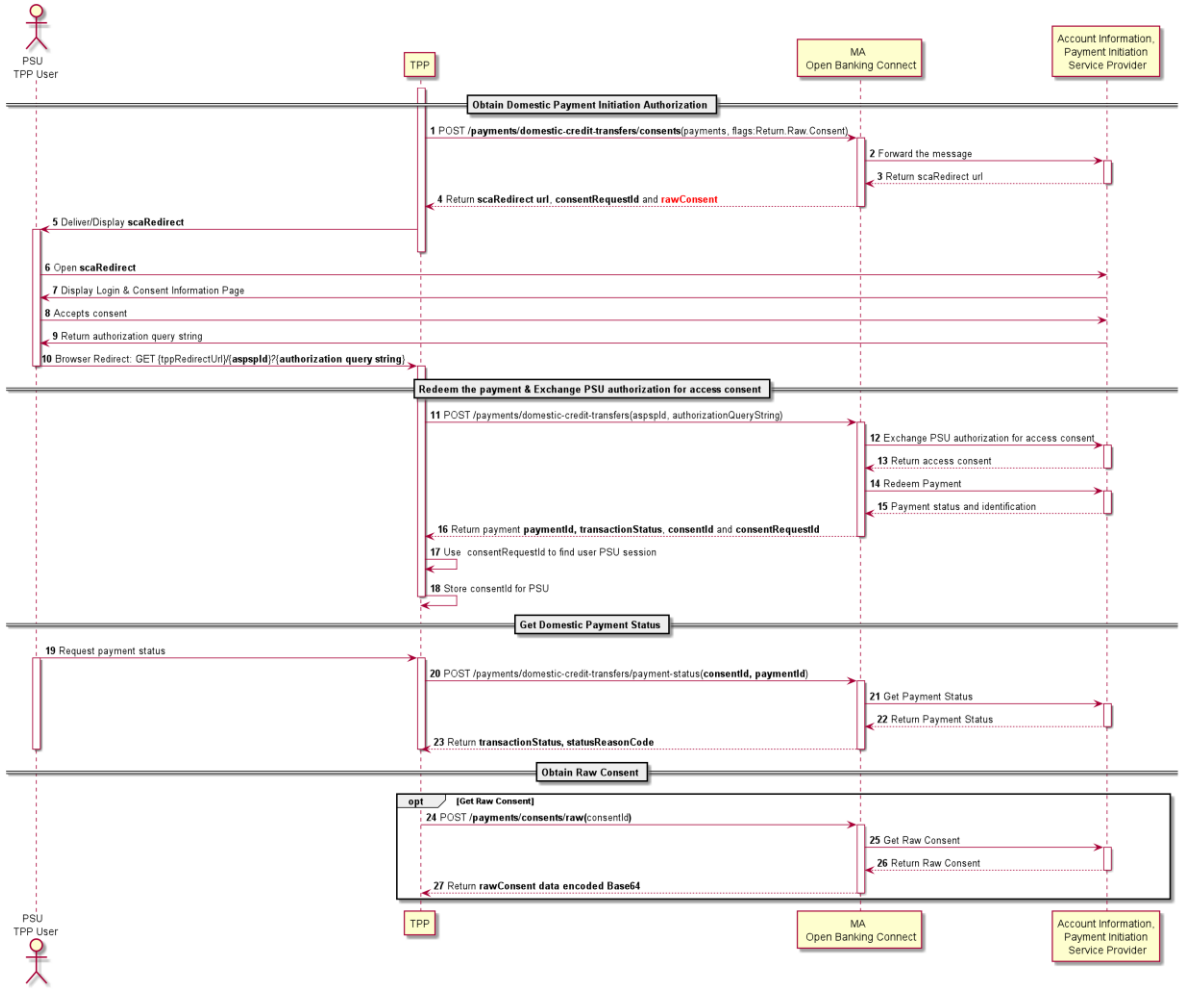
Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ signatureStatus	When TPP sends a request to a CMA9 API standard ASPSP, then MCOB will validate the signature in the authorization code. If the signature will pass validation, then the request will continue to Connectivity Partner and the response will include the successful status of signature validation, otherwise the request will be failed and TPP will receive back notification of failed signature validation	TPP	This element could be used to check TPP signature validation status	O	0..1	Enum	Status of validation of ASPSP's signature present in authorization code that TPP included in the request for credit transfer	Enum: "VALID", "UNKNOWN", "UNSIGNED", "NONCOMPLIANT"

For more information on errors received, see the error message structure described in Error Information.

8.4.3 Get Payment Status Request

The goal of this request is to retrieve from the ASPSP, based on the Consent ID previously obtained, the payment status for the specified Payment ID. This flow assumes the PSU has already consented to this access and the ASPSP system stores this record of consent or Consent ID. For more information, see section Payment Initiation Consent Request.

Image 35: Payment Information Service – Domestic - Get Payment Status Diagram



Request

Table 63 Get Payment Status Request

API Endpoint	POST /payments/domestic-credit-transfers/payment-status
API Profile	CMA9, Polish API

Request header

N/A

Request body

Table 64 Get Payment Status Request Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
requestInfo	Includes information about request being processed	MC	This element encapsulates all request information sent to the API Service.	M	1..1	Object	Set of elements used to define the request details	-
++ xRequestId	ID of the request, unique to the call, as determined by the initiating party (TPP)	TPP	A memorable ID could be used to support in a dispute	M	1..1	UUID	Free field that allows for the addition of information that can be referenced for future use.	36
++ consentId	ID of consent, unique to the call, as determinate by ASPSP after PSU authorization	ASPSP	Unique identification as assigned to identify the account access consent resource.	M	1..1	String	Consent ID received from ASPSP after consent authorisation	1-256

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ aspspId	ID of a financial institution servicing the Accounts of the PSU	ASPSP	Identification of ASPSP	M	1..1	UUID	Specifies the identification code of a financial institution which holds PSU accounts	36
++ merchant	Collect merchant data for reporting purposes	MC	Merchant data used for reporting or reconciliation purposes	O	0..1	Object	Set of elements used to define the merchant details	-
++++ id	Merchant identification code to identify the merchant	MC	Unique Merchant identifier per TPP, which could be used for reporting or reconciliation purposes	M	1..1	String	Field is available for TPPs to enable capturing of a merchant ID	1-256
++++ name	Merchant name to identify the merchant	MC	Name of merchant	M	1..1	String	Field is available for TPPs to enable capturing of a merchant name	1-256
paymentId	Unique identification as assigned by the ASPSP to uniquely identify the payment resource.	TPP	This value is used to make subsequent payment specific requests	M	1..1	String	Payment identification	1-256

Response – Success

HTTP Response Code = 200, OK

Note For a full list of response codes, including those returned in failure scenarios, see section Response Codes.

Response header

N/A

Response body

Table 65 Get Payment Status Request Response Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
originalRequestInfo	Returns original request information to the initiating party (TPP).	MC	This object contains original request information	M	1..1	Object	Original request information received from the initiating party	-
++ xRequestId	A memorable ID which could be used to support in a dispute	TPP	This element could be used for request-response tracking	M	1..1	UUID	Original xRequestId given by the client on request	36
payments	Specifies payments information in returned data	TPP	This element encapsulates all payments information	M	1..1	Object	Set of elements used to reference a payment instruction	-
++ transactionStatus	Specifies the status of the payment information group	TPP	Transaction Status of the payment resource	M	1.1	String	Transaction status	One of the following ISO 20022 statuses: • ACSC • ACSP • PDNG • RCVD • RJCT • CANC

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ statusReasonCode	Provides detailed information on the status reason.	TPP	Can only be used in case the status is equal to "RJCT"	O	0..1	String	Status Reason Code is used to add more information when a generic status code is returned by transaction Status field.	The codes for statusReasonCode are defined in ISO20022 - "ExternalStatusReason1Code". The following codes could be returned when RJCT status is raised: • AM04 • CURR • AM12 • DS24 • DS02 • AG08

For more information on errors received, see the error message structure described in Error Information.

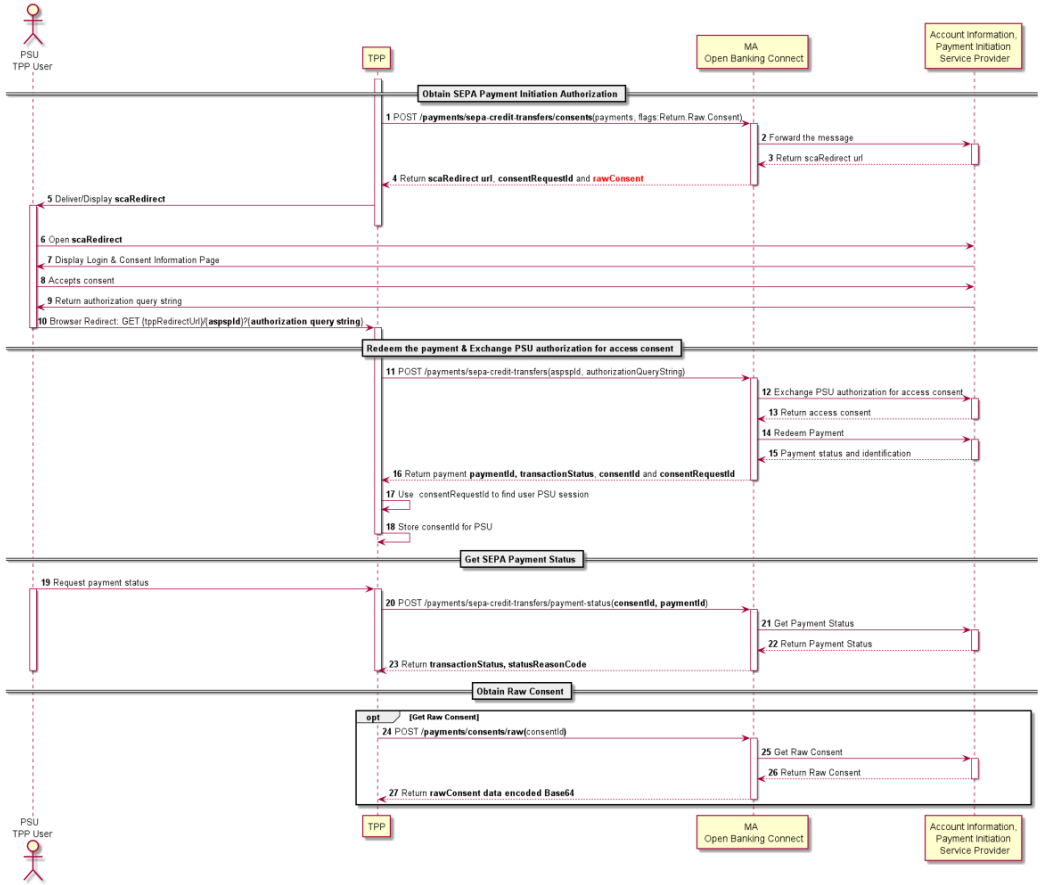
8.5 SEPA Payments

8.5.1 Payment Initiation Consent Request

The goal of this request is to create a payment initiation consent request that is authorised by the PSU.

Note SEPA payments are **NOT** supported by the CMA9 standard.

Image 36: Payment Information Service – SEPA - Payment Initiation Consent Diagram



Request

Table 66 Payment Initiation Consent Request

API Endpoint	POST /payments/sepa-credit-transfers/consents
API Profile	Polish API, NextGenPSD2, STET

Request header

N/A

Request body

Table 67 Payment Initiation Consent Request Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
requestInfo	Includes information about request being processed	MC	This element encapsulates all request information sent to the API Service	M	1..1	Object	Set of elements used to define the request details	-
++ xRequestId	ID of the request, unique to the call, as determined by the initiating party (TPP)	TPP	A memorable ID could be used to support in a dispute	M	1..1	UUID	Free field that allows for the addition of information that can be referenced for future use.	36

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ tppRedirectURI	URI of the TPP, where the transaction flow shall be redirected to after a Redirect	ASPSP	This element is used to specify the URI where should be redirected PSU after consent authorization	M	1..1	String	Call back URI where to redirect the PSU after authorisation of the consent	1-*
++ aspspId	ID of a financial institution servicing the Accounts of the PSU	ASPSP	Identification of ASPSP	M	1..1	UUID	This element is used to specify the identification code of a financial institution which holds PSU accounts	36
++ merchant	Collect merchant data for reporting purposes	MC	Merchant data used for reporting or reconciliation purposes	O	0..1	Object	Set of elements used to define the merchant details	-
++++ id	Merchant identification code to identify the merchant	MC	Unique Merchant identifier per TPP, which could be used for reporting or reconciliation purposes	M	1..1	String	Field is available for TPPs to enable capturing of a merchant ID	1-256
++++ name	Merchant name to identify the merchant	MC	Name of merchant	M	1..1	String	Field is available for TPPs to enable capturing of a merchant name	1-256
++ flags	The list of flags which can set the additional parameters in returned data	MC	This element is used to specify additional parameters which influence data returned in response	O	0..1	Enum	Request information flags which can influence the behaviour or returned data	Enum: ['Return.Raw.Consent']

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
payments	The object contains the list of fields used to send a payment instruction to financial institution	MC	This element encapsulates all payment fields	M	1..1	Object	Set of elements used to define the payments details	-
++ endToEndIdentification	Unique identification assigned by the initiating party to unambiguously identify the transaction. This identification is passed on, unchanged, throughout the entire end-to-end chain.	ASPSP	The end-to-end identification can be used for reconciliation or to link tasks relating to the transaction. It can be included in several messages related to the transaction.	M	1..1	String	Field is available for TPPs to enable end-to-end identification of the transaction.	1-18
++ localInstrument	User community specific instrument	ASPSP	This element is used to specify a local instrument, local clearing option and/or further qualify the service or service level.	M	1..1	Enum	User community specific instrument specifies which clearing system should be used to process the payment instruction	Enum: ["SEPA"]
++ instructionPriority	Indicator of the urgency or order of importance that the instructing party would like the instructed party to apply to the processing of the instruction	ASPSP	This element is used to specify urgency mode of the payment.	M	1..1	Enum	The way the transfer should be settled	Enum: ['Normal', 'Urgent']

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ categoryPurpose	Specifies the high-level purpose of the instruction based on a set of pre-defined categories	ASPSP	This is used by the initiating party to provide information concerning the processing of the payment	O	0..1	Enum	Category purpose	Enum: ['CASH', 'CORT', 'DVPM', 'INTC', 'TREA']
++ debtorAccount	Unambiguous identification of the account of the debtor to which a debit entry will be made as a result of the transaction	ASPSP	Specify account in a financial institution servicing the debtor	O	0..1	Object	Set of elements used to define the debtor account details	-
++++ iban	International Bank Account Number (IBAN) - identification used internationally by financial institutions to uniquely identify the account of a customer	ASPSP	This element is used to specify account identification number	M	1..1	String	Identification of the debtor account.	Only IBAN code is allowed.
++ debtorAgent	Unambiguous identification of the financial institution of the debtor to which a debit entry will be made as a result of the transaction	ASPSP	Specify financial institution servicing an account for the debtor	O	0..1	Object	Set of elements used to define the debtor agent details	-
++++ bicfi	When transferring money across international lines nearly always needs to use a SWIFT/BIC code, since it's the way banks and money transfer services identify where to send money on a global scale.	ASPSP	The bank SWIFT/BIC code is used to identify a specific bank in an international transaction	M	1..1	String	Identification of the debtor bank (BIC/SWIFT code).	1-256

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ debtorName	Name by which a party is known, and which is usually used to identify that party.	ASPSP	This element is used to specify debtor name	M	1..1	String	Debtor name	1-256
++ debtorAddress	Information that locates and identifies a specific address, as defined by postal services.	ASPSP	This element is usually used to identify the creditor party address.	O	0..1	Object	Set of elements used to define the debtor address details	-
++++ street	Name of a street or thoroughfare	ASPSP	This element is used to specify the street name	O	0..1	String	Street name	1-70
++++ buildingNumber	Number that identifies the position of a building on a street.	ASPSP	This element is used to specify the building number	O	0..1	String	Building number	1-256
++++ city	Name of a built-up area, with defined boundaries, and a local government.	ASPSP	This element is used to specify the city name	O	0..1	String	City name	1-256
++++ postalCode	Identifier consisting of a group of letters and/or numbers that is added to a postal address to assist the sorting of mail.	ASPSP	This element is used to specify the post code	O	0..1	String	Postal code	1-256
++++ countrySubDivision	Identifies a subdivision of a country such as state, region, county.	ASPSP	This element is used to specify the county code	O	0..1	String	Country sub-division	1-256
++++ country	Country in which a person resides	ASPSP	This element is used to specify the country code	O	1..1	String	Country name	ISO 3166 ALPHA 2 CHAR

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ instructedAmount	Amount of money to be moved between the debtor and creditor, before deduction of charges, expressed in the currency as ordered by the initiating party.	ASPSP	This amount has to be transported unchanged through the transaction chain	M	1..1	Object	Structure aiming to embed the amount and the currency to be used	-
++++ currency	A code allocated to a currency by a Maintenance Agency under an international identification scheme, as described in the latest edition of the international standard ISO 4217 "Codes for the representation of currencies and funds"	ASPSP	The currency of the amount used in transaction	M	1..1	Enum	Specifies the currency of the amount or of the account.	Enum: ["EUR"] Only EUR is allowed
++++ amount	A number of monetary units specified in an active currency where the unit of currency is explicit and compliant with ISO 4217.	ASPSP	This element is used to specify the amount of money to be processed	M	1..1	Decimal	Amount of money to be moved between the debtor and creditor, before deduction of charges, expressed in the currency as ordered by the initiating party	Double, min: 0
++ creditorAccount	Unambiguous identification of the account of the creditor to which a credit entry will be made as a result of the transaction	ASPSP	Specify account in a financial institution servicing the creditor	M	1..1	Object	Set of elements used to define the creditor account details	-

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++++ iban	International Bank Account Number (IBAN) - identification used internationally by financial institutions to uniquely identify the account of a customer	ASPSP	This element is used to specify account identification number	M	1..1	String	Identification of the creditor account.	Only IBAN is allowed.
++ creditorAgent	Unambiguous identification of the financial institution of the creditor to which a debit entry will be made as a result of the transaction	ASPSP	Specify financial institution servicing an account for the debtor	M	1..1	Object	Set of elements used to define the creditor agent details	-
++++ bicfi	When transferring money across international lines nearly always needs to use a SWIFT/BIC code, since it's the way banks and money transfer services identify where to send money on a global scale.	ASPSP	The bank SWIFT/BIC code is used to identify a specific bank in an international transaction	M	1..1	String	Identification of the creditor bank (BIC/SWIFT code).	1-256
++ creditorName	Name by which a party is known, and which is usually used to identify that party.	ASPSP	This element is used to specify debtor name	M	1..1	String	Creditor name	1-70
++ creditorAddress	Information that locates and identifies a specific address, as defined by postal services.	ASPSP	This element is usually used to identify the creditor party address.	M	1..1	object	Set of elements used to define the creditor address details	-
++++ street	Name of a street or thoroughfare	ASPSP	This element is used to specify the street name	O	0..1	String	Street name	1-70

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++++ buildingNumber	Number that identifies the position of a building on a street.	ASPSP	This element is used to specify the building number	O	0..1	String	Building number	1-256
++++ city	Name of a built-up area, with defined boundaries, and a local government.	ASPSP	This element is used to specify the city name	O	0..1	String	City name	1-256
++++ postalCode	Identifier consisting of a group of letters and/or numbers that is added to a postal address to assist the sorting of mail.	ASPSP	This element is used to specify the post code	O	0..1	String	Postal code	1-256
++++ countrySubDivision	Identifies a subdivision of a country such as state, region, county.	ASPSP	This element is used to specify the county code	O	0..1	String	Country sub-division	1-256
++++ country	Country in which a person resides	ASPSP	This element is used to specify the country code	M	1..1	String	Country name	ISO 3166 ALPHA 2 CHAR
++ remittanceInformationUnstructured	Information supplied to enable the matching/reconciliation of an entry with the items that the payment is intended to settle, such as commercial invoices in an accounts' receivable system, in an unstructured form.	ASPSP	Additional information regarding credit transfer	C	0..1	String	If available, the initiating party should provide this information in the unstructured remittance form	1-140

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ remittanceInformationStructured	Information supplied to enable the matching/reconciliation of an entry with the items that the payment is intended to settle	ASPSP	Additional information regarding credit transfer	O	0..1	String	If available, the initiating party should provide this information in the structured remittance form	1-256
++ requestedExecutionDate	Date at which the initiating party requests the clearing agent to process the payment.	ASPSP	This is the date on which the debtor's account is to be debited.	C	0..1	ISODate	Date when the transfer is to be executed (It is used for Future Dated payments)	ISODate, YYYY-MM-DD The requestedExecutionDate field and schedule object are both optional and mutually exclusive fields
++ remittanceInformationReference	Reference information provided by the creditor to allow the identification of the underlying documents	ASPSP	If available, the initiating party should provide this reference in the structured remittance information, to enable reconciliation by the creditor upon receipt of the amount of money.	O	0..1	String	Reference information, as assigned by the creditor, to unambiguously refer to the payment transaction.	1-256
++ schedule	Schedule object is used for standing orders.	ASPSP	Can be used to initiate a domestic standing orders payment instruction	C	0..1	Object	Set of elements used to define the schedule details	The schedule object and requestedExecutionDate field are both optional and mutually exclusive fields

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++++ startDate	The first date of execution for a standing order.	ASPSP	This element is used to specify the first payment date	M	1..1	ISODate	Payment first execution date	ISODate, YYYY-MM-DD
++++ frequency	Frequency rule for standing orders.	ASPSP	This element is used to specify the frequency of a payment	M	1..1	Enum	Payment frequency - determines how often a recurring payment will be made.	Enum: • DAIL - daily • WEEK - weekly • MNTH - monthly • QUTR - quarterly • SEMI - semi-annually • YEAR - yearly
++++ endDate	The last applicable day of execution for a given standing order. If not given, the standing order is considered as endless, until cancelled by PSU.	ASPSP	This element is used to specify payment last execution date	O	0..1	ISODate	Payment last execution date	ISODate, YYYY-MM-DD

Response – Success

HTTP Response Code = 200, OK

Note For a full list of response codes, including those returned in failure scenarios, see section Response Codes.

Response header

N/A

Response body

Table 68 Payment Initiation Consent Request Response Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
originalRequestInfo	Returns original request information to the initiating party (TPP).	MC	This object contains original request information	M	1..1	object	Original request information received from the initiating party	-
++ xRequestId	A memorable ID which could be used to support in a dispute	TPP	This element could be used for request-response tracking	M	1..1	UUID	Original xRequestId given by the client on request	36
aspspSCAApproach	Type of links admitted in this response	MC	This element is used to specify the SCA link types	O	0..1	Enum	Strong Customer Authentication (SCA) Approach	Enum: ["REDIRECT"]
consentRequestId	Unique identification as assigned by the TPP to uniquely identify the consent request.	TPP	This element could be used for consent request-response tracking	O	0..1	String	ID of the consent request	1-256
_links	The list of link types admitted in response, (further links might be added for ASPSP defined extensions)	MC	The actual hyperlinks used in the response depend on the dynamical decisions of the ASPSP when processing the request.	O	0..1	object	Set of elements used to define the links details	-
++ scaRedirect	In case of an SCA Redirect Approach, the ASPSP is transmitting the link to which to redirect the PSU browser	TPP	This element is used to specify the URL where the PSU should be redirected for consent authorization	M	1..1	URL	The URL where to redirect the PSU for authentication and authorization of the consent	1-*
additionalData	Specifies additional parameters in returned data	TPP	This element is used to specify additional parameters returned in response	C	0..1	Object	Set of elements used to define the additional data details	-

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ rawConsent	Returns raw consent data from associated ASPSP	TPP	Raw consent data could be used to extract additional information sent by ASPSP	M	1..1	String	Raw consent data received from ASPSP and encoded Base64.	1-* Returned only if requestInfo.flags contained 'Return.Raw.Consent' value

For more information on errors received, see the error message structure described in Error Information.

After the response is received by a TPP, it will have to redirect the PSU to the URI received in 'scaRedirect' to complete the authentication with the ASPSP and authorisation of the consent.

After completing this process, the TPP will obtain an access token (Consent Id) on the URI address specified in the 'tppRedirectURI' field in the request. Depending of the ASPSP profile, the TPP can also get additional parameters along with access token. The following table shows the list of parameters which a TPP may get from an ASPSP.

Table 69 ASPSPs parameters list

Name	Type	API Profile	Description
memberId	String	All Profiles	Member identification
code	String	All Profiles	Authorization string
state	String	All Profiles	State data
tokenId/consentId	String	All Profiles	Access token
transferId	String	STET	Transfer identification
transferStatus	String	STET	Transfer status

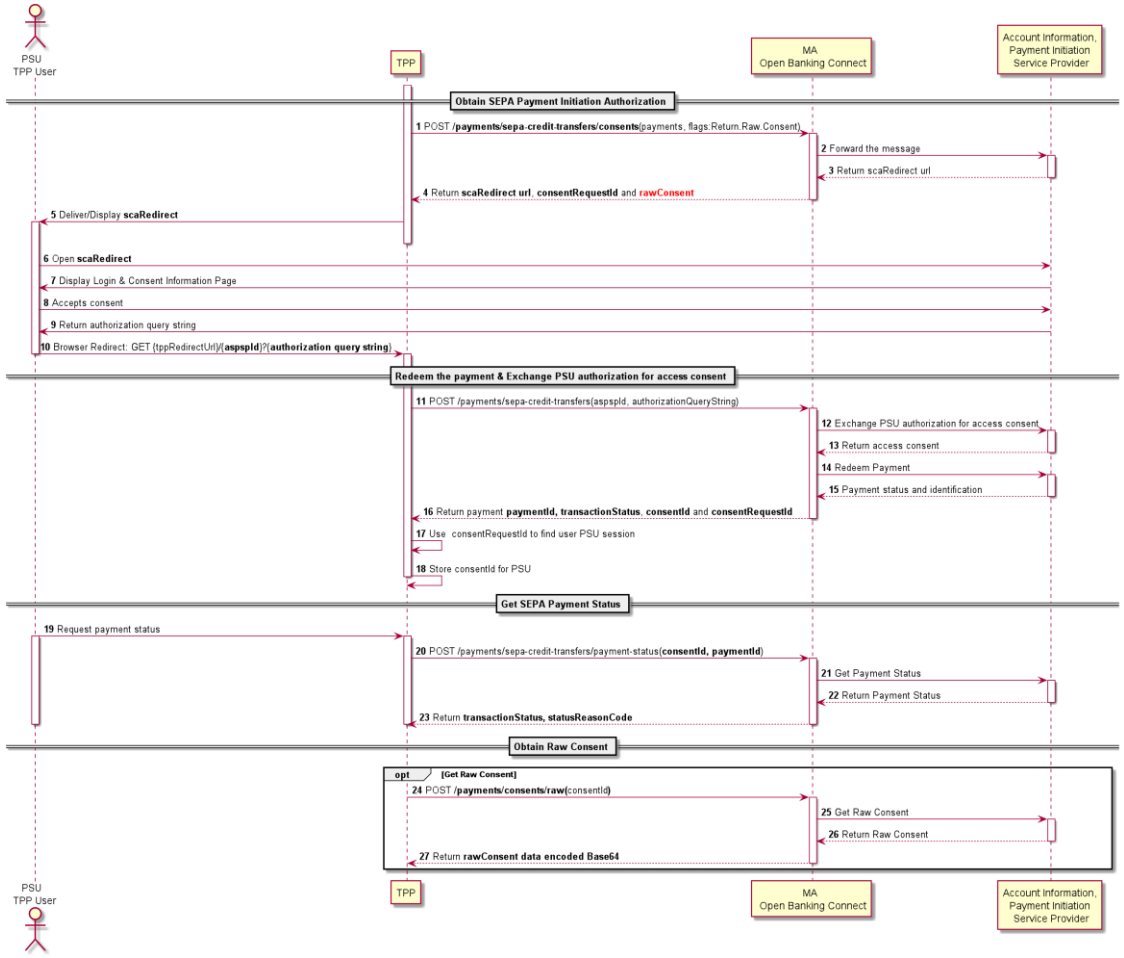
Note The parameters sent from ASPSP to TPP are Base64 encoded.

The ASPSPs using the STET profile do not need to redeem the token separately (i.e. call Payment Credit Transfer Initiation endpoint). This is because the token is redeemed during the consent request operation and the transfer Id and transfer status is returned in the URI address specified in the 'tppRedirectURI' field of the request.

8.5.2 Payment Credit Transfer Initiation Request

The goal of this request is to create a payment initiation request on the ASPSP side using a consent authorised by the PSU. This flow assumes the PSU has already consented to this access and the ASPSP system stores this record of consent or Consent ID. For more information, see section Payment Initiation Consent Request.

Image 37: Payment Information Service – SEPA - Payment Credit Transfer Initiation Diagram



Request

Table 70 Payment Credit Transfer Initiation Request

API Endpoint	POST /payments/sepa-credit-transfers
API Profile	Polish API, NextGenPSD2, STET

Request header

N/A

Request body

Table 71 Payment Credit Transfer Initiation Request Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
requestInfo	Includes information about request being processed	MC	This element encapsulates all request information sent to the API Service.	M	1..1	Object	Set of elements used to define the request details	-
++ xRequestId	ID of the request, unique to the call, as determined by the initiating party (TPP)	TPP	A memorable ID could be used to support in a dispute	M	1..1	UUID	Free field that allows for the addition of information that can be referenced for future use.	36
++ aspspId	ID of a financial institution servicing the Accounts of the PSU	ASPSP	Identification of ASPSP	M	1..1	UUID	Specifies the identification code of a financial institution which holds PSU accounts	36
++ merchant	Collect merchant data for reporting purposes	MC	Merchant data used for reporting or reconciliation purposes	O	0..1	Object	Set of elements used to define the merchant details	-

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++++ id	Merchant identification code to identify the merchant	MC	Unique Merchant identifier per TPP, which could be used for reporting or reconciliation purposes	M	1..1	String	Field is available for TPPs to enable capturing of a merchant ID	1-256
++++ name	Merchant name to identify the merchant	MC	Name of merchant	M	1..1	String	Field is available for TPPs to enable capturing of a merchant name	1-256
++ authorization	Authorization data received from associated ASPSP	MC	Authorization data is used to exchange it for access consent	M	1..1	String	The authorization data received after PSU has authorized the consent	1-*

Response – Success

HTTP Response Code = 200, OK

Note For a full list of response codes, including those returned in failure scenarios, see section Response Codes.

Response header

N/A

Response body

Table 72 Payment Credit Transfer Initiation Request Response Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
originalRequestInfo	Returns original request information to the initiating party (TPP).	MC	This object contains original request information	M	1..1	object	Original request information received from the initiating party	-
++ xRequestId	A memorable ID which could be used to support in a dispute	TPP	This element could be used for request-response tracking	M	1..1	UUID	Original xRequestId given by the client on request	36
transfer	Specifies transfer information in returned data	TPP	This element encapsulates all transfer information	M	1..1	Object	Set of elements used to reference a payment instruction	-
++ paymentId	Unique identification as assigned by the ASPSP to uniquely identify the payment resource.	TPP	This value should be used to make subsequent payment specific requests	M	1..1	String	PI / Transfer identification	1-256
++ transactionStatus	Specifies the status of the payment information group	TPP	Transaction Status of the payment resource	M	1..1	Enum	Status of the payment	Enum: • ACSC • ACSP • PDNG • RCVD • RJCT • CANC

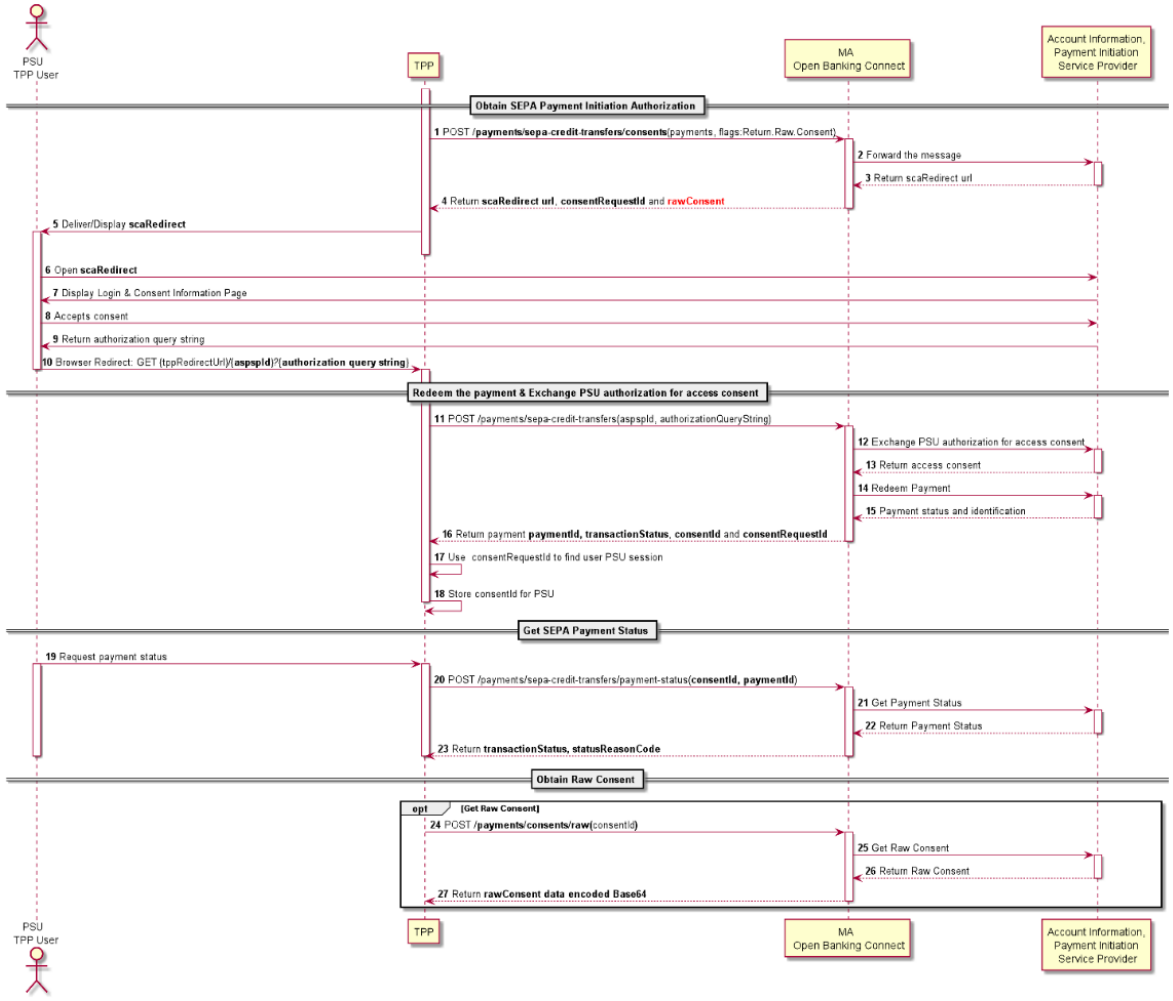
Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ statusReasonCode	Provides detailed information on the status reason.	TPP	Can only be used in case the status is equal to "RJCT"	O	0..1	String	Status Reason Code is used to add more information when a generic status code is returned by transaction Status field.	The codes for statusReasonCode are defined in ISO20022 - "ExternalStatusReason1Code". The following codes could be returned when RJCT status is raised: • AM04 • CURR • AM12 • DS24 • DS02 • AG08
consent	Specifies consent information in returned data	TPP	This element encapsulates all consent information	O	0..1	Object	Set of elements used to define the consent details	-
++ consentId	ID of consent, unique to the call, as determinate by ASPSP after PSU authorization	TPP	Unique identification as assigned to identify the account access consent resource.	O	0..1	String	Access token (consent Id) received from ASPSP after consent authorisation	1-256
++ consentRequestId	Unique identification as assigned by the TPP to uniquely identify the consent request.	TPP	This element could be used for consent request-response tracking	O	0..1	String	Request consent identification	1-256

For more information on errors received, see the error message structure described in Error Information.

8.5.3 Get Payment Status Request

The goal of this request is to check the status of a previously made SEPA payment. This flow assumes the PSU has already consented to this access and the ASPSP system stores this record of consent or Consent ID. For more information, see section Payment Initiation Consent Request.

Image 38: Payment Information Service – SEPA - Get Payment Status Diagram



Request

Table 73 Get Payment Status Request

API Endpoint	POST /payments/sepa-credit-transfers/payment-status
API Profile	Polish API, NextGenPSD2, STET

Request header

N/A

Request body

Table 74 Get Payment Status Request Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
requestInfo	Includes information about request being processed	MC	This element encapsulates all request information sent to the API Service.	M	1..1	Object	Set of elements used to define the request details	-
++ xRequestId	ID of the request, unique to the call, as determined by the initiating party (TPP)	TPP	A memorable ID could be used to support in a dispute	M	1..1	UUID	Free field that allows for the addition of information that can be referenced for future use.	36

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ consentId	ID of consent, unique to the call, as determinate by ASPSP after PSU authorization	ASPSP	Unique identification as assigned to identify the account access consent resource.	M	1..1	String	Consent ID received from ASPSP after consent authorisation	1-256
++ aspspId	ID of a financial institution servicing the Accounts of the PSU	ASPSP	Identification of ASPSP	M	1..1	UUID	Specifies the identification code of a financial institution which holds PSU accounts	36
++ merchant	Collect merchant data for reporting purposes	MC	Merchant data used for reporting or reconciliation purposes	O	0..1	Object	Set of elements used to define the merchant details	-
++++ id	Merchant identification code to identify the merchant	MC	Unique Merchant identifier per TPP, which could be used for reporting or reconciliation purposes	M	1..1	String	Field is available for TPPs to enable capturing of a merchant ID	1-256
++++ name	Merchant name to identify the merchant	MC	Name of merchant	M	1..1	String	Field is available for TPPs to enable capturing of a merchant name	1-256
paymentId	Unique identification as assigned by the ASPSP to uniquely identify the payment resource.	TPP	This value is used to make subsequent payment specific requests	M	1..1	String	PI / Transfer identification	1-256

Response – Success

HTTP Response Code = 200, OK

Note For a full list of response codes, including those returned in failure scenarios, see section Response Codes.

Response header

N/A

Response body

Table 75 Get Payment Status Request Response Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
originalRequestInfo	Returns original request information to the initiating party (TPP).	MC	This object contains original request information	M	1..1	Object	Original request information received from the initiating party	-
++ xRequestId	A memorable ID which could be used to support in a dispute	TPP	This element could be used for request-response tracking	M	1..1	UUID	Original xRequestId given by the client on request	36
payments	Specifies payments information in returned data	TPP	This element encapsulates all payments information	C	0..1	Object	Set of elements used to reference a payment instruction	-

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ transactionStatus	Specifies the status of the payment information group	TPP	Transaction Status of the payment resource	M	1.1	String	Status of the payment	One of the following ISO 20022 statuses: • ACSC • ACSP • PDNG • RCVD • RJCT • CANC
++ statusReasonCode	Provides detailed information on the status reason.	TPP	Can only be used in case the status is equal to "RJCT"	O	0..1	String	Status Reason Code is used to add more information when a generic status code is returned by transaction Status field.	The codes for statusReasonCode are defined in ISO20022 - "ExternalStatusReason 1Code". The following codes could be returned when RJCT status is raised: • AM04 • CURR • AM12 • DS24 • DS02 • AG08

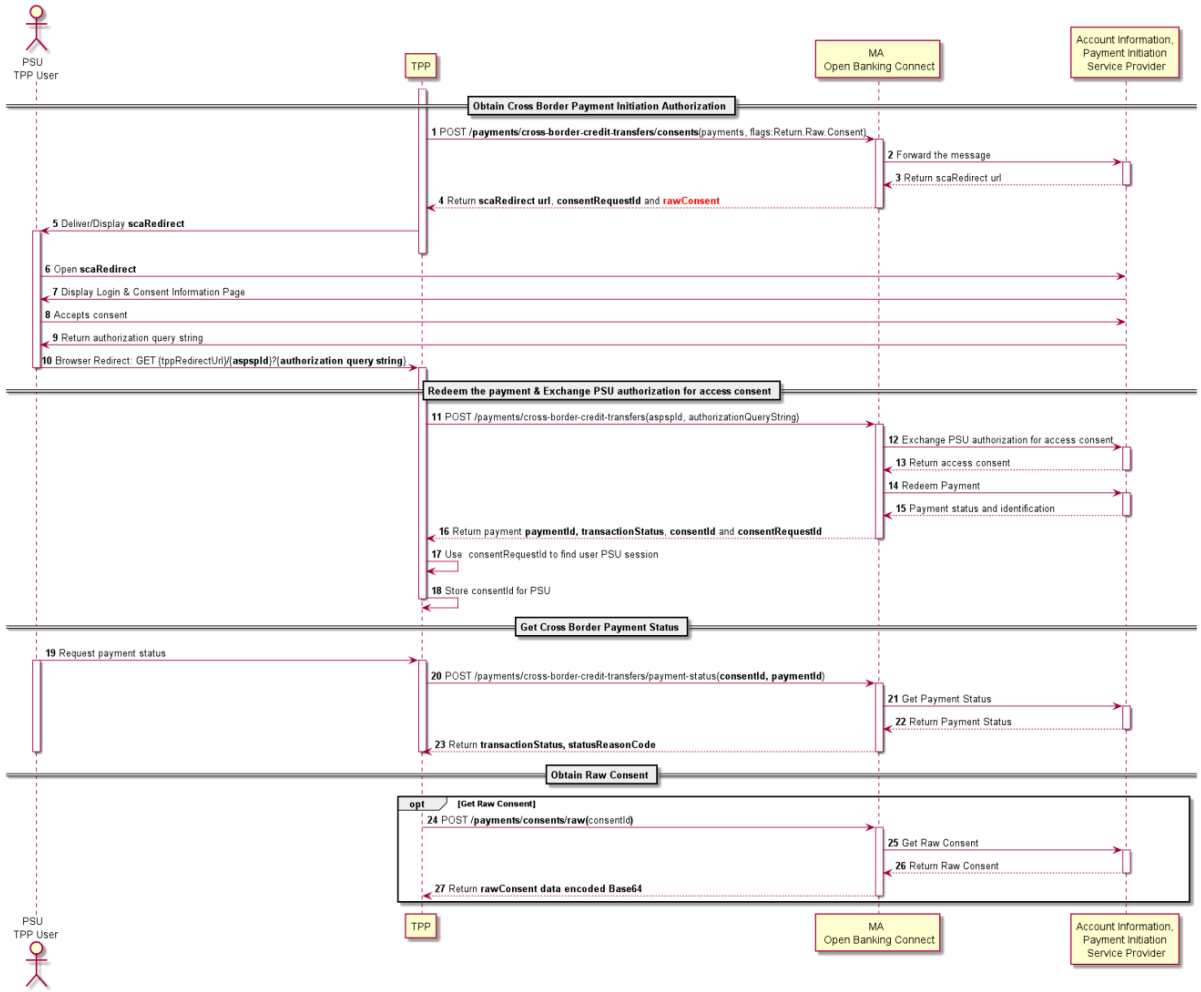
For more information on errors received, see the error message structure described in Error Information.

8.6 *Cross-Border (International) Payments*

8.6.1 Payment Initiation Consent Request

The goal of this request is to create a payment initiation consent request that is authorised by the PSU.

Image 39: Payment Information Service – Cross-Border - Payment Initiation Consent Diagram



Request

Table 76 Payment Initiation Consent Request

API Endpoint	POST /payments/cross-border-credit-transfers/consents
API Profile	CMA9, Polish API

Request header

N/A

Request body

Table 77 Payment Initiation Consent Request Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
requestInfo	Includes information about request being processed	MC	This element encapsulates all request information sent to the API Service	M	1..1	Object	Set of elements used to define the request details	-
++ xRequestId	ID of the request, unique to the call, as determined by the initiating party (TPP)	TPP	A memorable ID could be used to support in a dispute	M	1..1	UUID	Free field that allows for the addition of information that can be referenced for future use.	36
++ tppRedirectURI	URI of the TPP, where the transaction flow shall be redirected to after a Redirect	ASPSP	This element is used to specify the URI where should be redirected PSU after consent authorization	M	1..1	String	Call back URI where to redirect the PSU after authorisation of the consent	1-*

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ aspspId	ID of a financial institution servicing the Accounts of the PSU	ASPSP	Identification of ASPSP	M	1..1	UUID	This element is used to specify the identification code of a financial institution which holds PSU accounts	36
++ merchant	Collect merchant data for reporting purposes	MC	Merchant data used for reporting or reconciliation purposes	O	0..1	Object	Set of elements used to define the merchant details	-
++++ id	Merchant identification code to identify the merchant	MC	Unique Merchant identifier per TPP, which could be used for reporting or reconciliation purposes	M	1..1	String	Field is available for TPPs to enable capturing of a merchant ID	1-256
++++ name	Merchant name to identify the merchant	MC	Name of merchant	M	1..1	String	Field is available for TPPs to enable capturing of a merchant name	1-256
++ flags	The list of flags which can set the additional parameters in returned data	MC	This element is used to specify additional parameters which influence data returned in response	O	0..1	Enum	Request information flags which can influence the behaviour or returned data	Enum: ['Return.Raw.Consent']
payments	The object contains the list of fields used to send a payment instruction to financial institution	MC	This element encapsulates all payment fields	M	1..1	Object	Set of elements used to define the payments details	-

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ endToEndIdentification	Unique identification assigned by the initiating party to unambiguously identify the transaction. This identification is passed on, unchanged, throughout the entire end-to-end chain.	ASPSP	The end-to-end identification can be used for reconciliation or to link tasks relating to the transaction. It can be included in several messages related to the transaction.	M	1..1	String	Field is available for TPPs to enable end-to-end identification of the transaction.	1-18
++ localInstrument	User community specific instrument	ASPSP	This element is used to specify a local instrument, local clearing option and/or further qualify the service or service level.	M	1..1	Enum	User community specific instrument specifies which clearing system should be used to process the payment instruction	Enum: ["SWIFT"]
++ instructionPriority	Indicator of the urgency or order of importance that the instructing party would like the instructed party to apply to the processing of the instruction	ASPSP	This element is used to specify urgency mode of the payment.	M	1..1	Enum	The way the transfer should be settled	Enum: ['Normal', 'Urgent']

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ debtorAccount	Unambiguous identification of the account of the debtor to which a debit entry will be made as a result of the transaction	ASPSP	Specify account in a financial institution servicing the debtor	M	1..1	Object	Set of elements used to define the debtor account details	-
++++ schemeName	Name of the identification scheme, in a coded form as published in an external list.	ASPSP	This field is used to identify the type of Identification used to identify an account.	M	1..1	Enum	Account scheme name 'SortCodeAccountNumber' = Sort Code + Account Number 'IBAN' = International Bank Account Number	Enum: ['SortCodeAccountNumber','IBAN']
++++ identification	Identification assigned by an institution to identify an account. This identification is known by the account owner.	ASPSP	This element is used to specify the account identification number	M	1..1	String	Account identification which can be used on payload-level to address specific accounts	1-256
++ debtorName	Name by which a party is known, and which is usually used to identify that party.	ASPSP	This element is used to specify debtor name	O	0..1	String	Debtor name	1-256
++ debtorAddress	Information that locates and identifies a specific address, as defined by postal services.	ASPSP	This element is usually used to identify the creditor party address.	O	0..1	object	Set of elements used to define the debtor address details	-

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++++ street	Name of a street or thoroughfare	ASPSP	This element is used to specify the street name	O	0..1	String	Street name	1-70
++++ buildingNumber	Number that identifies the position of a building on a street.	ASPSP	This element is used to specify the building number	O	0..1	String	Building number	1-256
++++ city	Name of a built-up area, with defined boundaries, and a local government.	ASPSP	This element is used to specify the city name	O	0..1	String	City name	1-256
++++ postalCode	Identifier consisting of a group of letters and/or numbers that is added to a postal address to assist the sorting of mail.	ASPSP	This element is used to specify the post code	O	0..1	String	Postal code	1-256
++++ countrySubDivision	Identifies a subdivision of a country such as state, region, county.	ASPSP	This element is used to specify the county code	O	0..1	String	Country sub-division	1-256
++++ country	Country in which a person resides	ASPSP	This element is used to specify the country code	O	1..1	String	Country name	ISO 3166 ALPHA 2 CHAR

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ instructedAmount	Amount of money to be moved between the debtor and creditor, before deduction of charges, expressed in the currency as ordered by the initiating party.	ASPSP	This amount has to be transported unchanged through the transaction chain	M	1..1	Object	Instructed Amount object	-
++++ currency	A code allocated to a currency by a Maintenance Agency under an international identification scheme, as described in the latest edition of the international standard ISO 4217 "Codes for the representation of currencies and funds"	ASPSP	The currency of the amount used in transaction	M	1..1	String	Currency code as per ISO 4217.	ISO 4217 ALPHA 3 CHAR
++++ amount	A number of monetary units specified in an active currency where the unit of currency is explicit and compliant with ISO 4217.	ASPSP	This element is used to specify the amount of money to be processed	M	1..1	Decimal	Amount value	Double, min: 0

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ creditorAccount	Unambiguous identification of the account of the creditor to which a credit entry will be posted as a result of the payment transaction.	ASPSP	Specify account in a financial institution servicing the creditor	M	1..1	Object	Set of elements used to define the creditor account details	-
++++ schemeName	Name of the identification scheme, in a coded form as published in an external list.	ASPSP	This field is used to identify the type of Identification used to identify an account.	M	1..1	Enum	Account scheme name 'SortCodeAccountNumber' = Sort Code + Account Number 'IBAN' = International Bank Account Number	Enum: ['SortCodeAccountNumber','IBAN']
++++ identification	Identification assigned by an institution to identify an account. This identification is known by the account owner.	ASPSP	This element is used to specify the account identification number	M	1..1	String	Account identification which can be used on payload-level to address specific accounts	1-256
++ creditorAgent	Unambiguous identification of the financial institution of the creditor to which a credit entry will be posted as a result of the payment transaction.	ASPSP	Specify financial institution servicing an account for the creditor.	M	1..1	Object	Set of elements used to define the creditor agent details	-
++++ schemeName	Name of the identification scheme, in a coded form as published in an external list.	ASPSP	This field is used to identify the type of Identification used to identify an account.	M	1..1	Enum	Agent scheme name 'BIC' = Bank International Code	Enum: 'BIC'

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++++ identification	Code allocated to a financial institution by the ISO 9362 Registration Authority to identify the financial institution	ASPSP	This element is used to specify the financial institution identification code	M	1..1	String	Identification code to identify the financial institution.	1-256
++ creditorName	Name by which a party is known, and which is usually used to identify that party.	ASPSP	This element is used to specify debtor name	M	1..1	String	Creditor name	1-70
++ creditorAddress	Information that locates and identifies a specific address, as defined by postal services.	ASPSP	This element is usually used to identify the creditor party address.	O	0..1	object	Creditor address object	-
++++ street	Name of a street or thoroughfare	ASPSP	This element is used to specify the street name	O	0..1	String	Street name	1-70
++++ buildingNumber	Number that identifies the position of a building on a street.	ASPSP	This element is used to specify the building number	O	0..1	String	Building number	1-256
++++ city	Name of a built-up area, with defined boundaries, and a local government.	ASPSP	This element is used to specify the city name	O	0..1	String	City name	1-256

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++++ postalCode	Identifier consisting of a group of letters and/or numbers that is added to a postal address to assist the sorting of mail.	ASPSP	This element is used to specify the post code	O	0..1	String	Postal code	1-256
++++ countrySubDivision	Identifies a subdivision of a country such as state, region, county.	ASPSP	This element is used to specify the county code	O	0..1	String	Country sub-division	1-256
++++ country	Country in which a person resides	ASPSP	This element is used to specify the country code	M	1..1	String	Country name	ISO 3166 ALPHA 2 CHAR
++ remittanceInformation Unstructured	Information supplied to enable the matching/reconciliation of an entry with the items that the payment is intended to settle, such as commercial invoices in an accounts' receivable system, in an unstructured form.	ASPSP	Additional information regarding credit transfer	C	0..1	String	If available, the initiating party should provide this information in the unstructured remittance form	1-140

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ requestedExecutionDate	Date at which the initiating party requests the clearing agent to process the payment.	ASPSP	This is the date on which the debtor's account is to be debited.	C	0..1	ISODate	Date when the transfer is to be executed (It is used for Future Dated payments)	ISODate, YYYY-MM-DD The requestedExecutionDate field and schedule object are both optional and mutually exclusive fields
++ transferCharges	Specifies which party/parties will bear the charges associated with the processing of the payment transaction.	ASPSP	The transferCharges field is used by the PISP to indicate the bearer of charges. An ASPSP could reject the request if the requested charge allocation cannot be fulfilled	O	0..1	Enum	Transfer charges code. SEN = Sender SHA = Shared BEN = Beneficiary	Enum: ["SEN", "SHA", "BEN"]
++ remittanceInformationReference	Reference information provided by the creditor to allow the identification of the underlying documents	ASPSP	If available, the initiating party should provide this reference in the structured remittance information, to enable reconciliation by the creditor upon receipt of the amount of money.	O	0..1	String	Reference information, as assigned by the creditor, to unambiguously refer to the payment transaction.	1-256

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ instructionIdentification	Unique identification as assigned by an instructing party for an instructed party to unambiguously identify the instruction.	ASPSP	The instruction identification is a point to point reference that can be used between the instructing party and the instructed party to refer to the individual instruction. It can be included in several messages related to the instruction.	O	0..1	String	Unique identification shared between the PISP and the ASPSP	1-256

Response – Success

HTTP Response Code = 200, OK

Note For a full list of response codes, including those returned in failure scenarios, see section Response Codes.

Response header

N/A

Response body

Table 78 Payment Initiation Consent Request Response Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
originalRequestInfo	Returns original request information to the initiating party (TPP).	MC	This object contains original request information	M	1..1	object	Original request information received from the initiating party	-
++ xRequestId	A memorable ID which could be used to support in a dispute	TPP	This element could be used for request-response tracking	M	1..1	UUID	Original xRequestId given by the client on request	36
aspspSCAApproach	Type of links admitted in this response	MC	This element is used to specify the SCA link types	O	0..1	Enum	Strong Customer Authentication (SCA) Approach	Enum: ["REDIRECT"]
consentRequestId	Unique identification as assigned by the TPP to uniquely identify the consent request.	TPP	This element could be used for consent request-response tracking	O	0..1	String	ID of the consent request	1-256
_links	The list of link types admitted in response, (further links might be added for ASPSP defined extensions)	MC	The actual hyperlinks used in the response depend on the dynamical decisions of the ASPSP when processing the request.	O	0..1	object	Set of elements used to define the links details	-

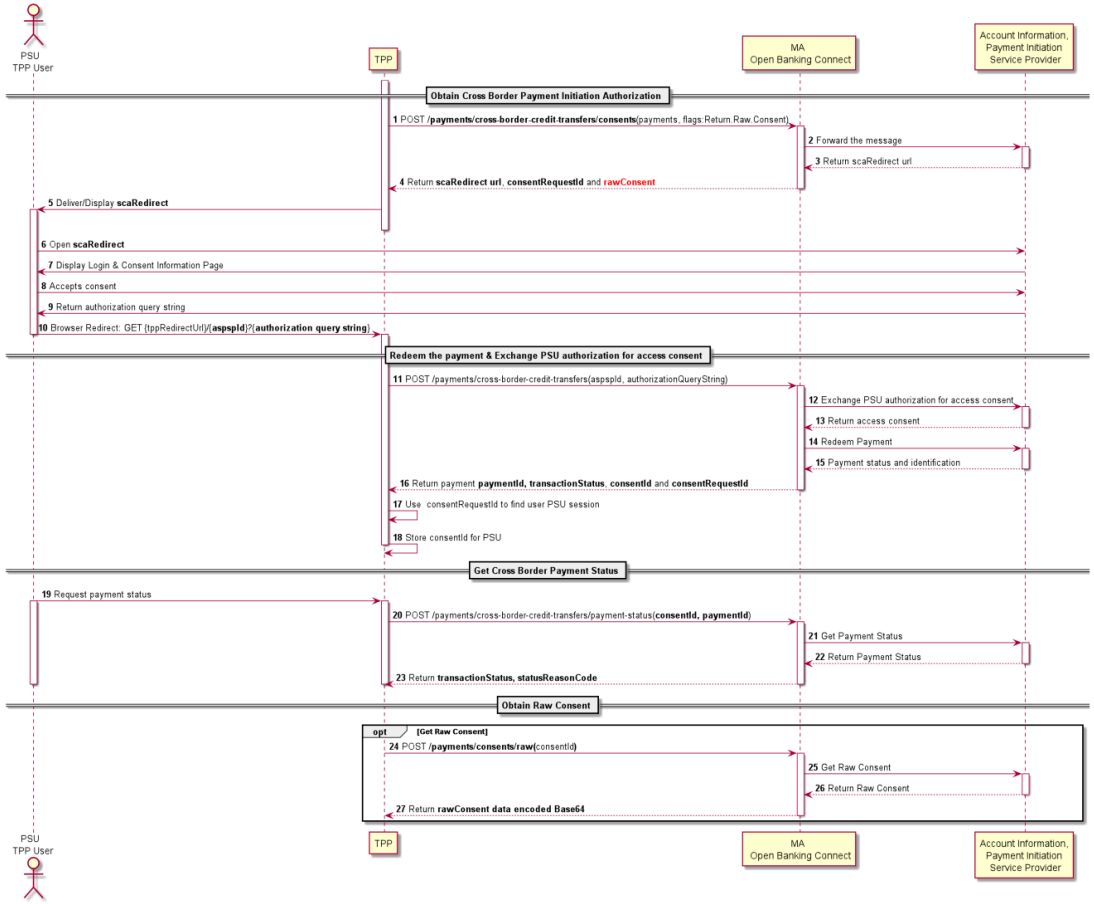
Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ scaRedirect	In case of an SCA Redirect Approach, the ASPSP is transmitting the link to which to redirect the PSU browser	TPP	This element is used to specify the URL where the PSU should be redirected for consent authorization	M	1..1	URL	The URL where to redirect the PSU for authentication and authorization of the consent	1-*
additionalData	Specifies additional parameters in returned data	TPP	This element is used to specify additional parameters returned in response	C	0..1	Object	Set of elements used to define the additional data details	-
++ rawConsent	Returns raw consent data from associated ASPSP	TPP	Raw consent data could be used to extract additional information sent by ASPSP	M	1..1	String	Raw consent data received from ASPSP and encoded Base64.	1-* Returned only if requestInfo.flags contained 'Return.Raw.Consent' value

For more information on errors received, see the error message structure described in Error Information.

8.6.2 Payment Credit Transfer Initiation Request

The goal of this request is to create a payment initiation request on the ASPSP side using a consent authorised by the PSU. This flow assumes the PSU has already consented to this access and the ASPSP system stores this record of consent or Consent ID. For more information, see section Payment Initiation Consent Request.

Image 40: Payment Information Service – Cross-Border - Payment Credit Transfer Initiation Diagram



Request

Table 79 Payment Credit Transfer Initiation Request

API Endpoint	POST /payments/cross-border-credit-transfers
API Profile	CMA9, Polish API

Request header

N/A

Request body

Table 80 Payment Credit Transfer Initiation Request Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
requestInfo	Includes information about request being processed	MC	This element encapsulates all request information sent to the API Service.	M	1..1	Object	Set of elements used to define the request details	-
++ xRequestId	ID of the request, unique to the call, as determined by the initiating party (TPP)	TPP	A memorable ID could be used to support in a dispute	M	1..1	UUID	Free field that allows for the addition of information that can be referenced for future use.	36
++ aspspId	ID of a financial institution servicing the Accounts of the PSU	ASPSP	Identification of ASPSP	M	1..1	UUID	Specifies the identification code of a financial institution which holds PSU accounts	36

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ merchant	Collect merchant data for reporting purposes	MC	Merchant data used for reporting or reconciliation purposes	O	0..1	Object	Set of elements used to define the merchant details	-
++++ id	Merchant identification code to identify the merchant	MC	Unique Merchant identifier per TPP, which could be used for reporting or reconciliation purposes	M	1..1	String	Field is available for TPPs to enable capturing of a merchant ID	1-256
++++ name	Merchant name to identify the merchant	MC	Name of merchant	M	1..1	String	Field is available for TPPs to enable capturing of a merchant name	1-256
++ authorization	Authorization data received from associated ASPSP	MC	Authorization data is used to exchange it for access consent	M	1..1	String	The authorization data received after PSU has authorized the consent	1-*

Response – Success

HTTP Response Code = 200, OK

Note For a full list of response codes, including those returned in failure scenarios, see section Response Codes.

Response header

N/A

Response body

Table 81 Payment Credit Transfer Initiation Request Response Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
originalRequestInfo	Returns original request information to the initiating party (TPP).	MC	This object contains original request information	M	1..1	object	Original request information received from the initiating party	-
++ xRequestId	A memorable ID which could be used to support in a dispute	TPP	This element could be used for request-response tracking	M	1..1	UUID	Original xRequestId given by the client on request	36
transfer	Specifies transfer information in returned data	TPP	This element encapsulates all transfer information	M	1..1	Object	Set of elements used to reference a payment instruction	-
++ paymentId	Unique identification as assigned by the ASPSP to uniquely identify the payment resource.	TPP	This value should be used to make subsequent payment specific requests	M	1..1	String	PI / Transfer identification	1-256
++ transactionStatus	Specifies the status of the payment information group	TPP	Transaction Status of the payment resource	M	1..1	Enum	Status of the payment	Enum: • ACSC • ACSP • PDNG • RCVD • RJCT • CANC

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ statusReasonCode	Provides detailed information on the status reason.	TPP	Can only be used in case the status is equal to "RJCT"	O	0..1	String	Status Reason Code is used to add more information when a generic status code is returned by transaction Status field.	The codes for statusReasonCode are defined in ISO20022 - "ExternalStatusReason1Code". The following codes could be returned when RJCT status is raised: • AM04 • CURR • AM12 • DS24 • DS02 • AG08
consent	Specifies consent information in returned data	TPP	This element encapsulates all consent information	O	0..1	Object	Set of elements used to define the consent details	-
++ consentId	ID of consent, unique to the call, as determinate by ASPSP after PSU authorization	TPP	Unique identification as assigned to identify the account access consent resource.	O	0..1	String	Access token (consent Id) received from ASPSP after consent authorisation	1-256
++ consentRequestId	Unique identification as assigned by the TPP to uniquely identify the consent request.	TPP	This element could be used for consent request- response tracking	O	0..1	String	Request consent identification	1-256

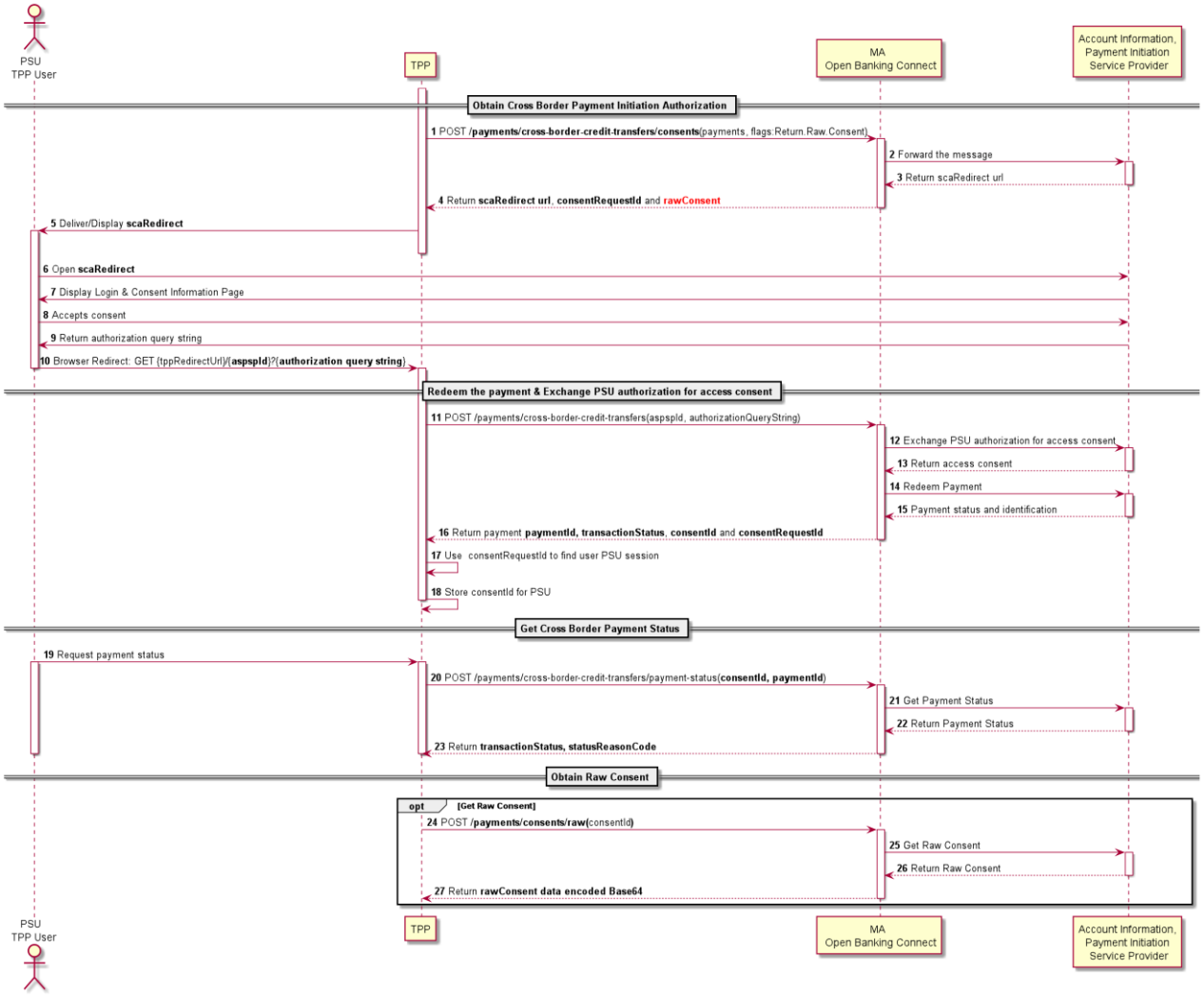
Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ signatureStatus	When TPP sends a request to a CMA9 API standard ASPSP, then MCOB will validate the signature in the authorization code. If the signature will pass validation, then the request will continue to Connectivity Partner and the response will include the successful status of signature validation, otherwise the request will be failed and TPP will receive back notification of failed signature validation	TPP	This element could be used to check TPP signature validation status	O	0..1	Enum	Status of validation of ASPSP's signature present in authorization code that TPP included in the request for credit transfer	Enum: "VALID", "UNKNOWN", "UNSIGNED", "NONCOMPLIANT "

For more information on errors received, see the error message structure described in Error Information.

8.6.3 Get Payment Status Request

The goal of this request is to check the status of a previously made Cross-Border payment. This flow assumes the PSU has already consented to this access and the ASPSP system stores this record of consent or Consent ID. For more information, see section Payment Initiation Consent Request.

Image 41: Payment Information Service – Cross-Border - Get Payment Status Diagram



Request

Table 82 Get Payment Status Request

API Endpoint	POST /payments/cross-border-credit-transfers /payment-status
API Profile	CMA9, Polish API

Request header

N/A

Request body

Table 83 Get Payment Status Request Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
requestInfo	Includes information about request being processed	MC	This element encapsulates all request information sent to the API Service.	M	1..1	Object	Set of elements used to define the request details	-
++ xRequestId	ID of the request, unique to the call, as determined by the initiating party (TPP)	TPP	A memorable ID could be used to support in a dispute	M	1..1	UUID	Free field that allows for the addition of information that can be referenced for future use.	36
++ consentId	ID of consent, unique to the call, as determinate by ASPSP after PSU authorization	ASPSP	Unique identification as assigned to identify the account access consent resource.	M	1..1	String	Consent ID received from ASPSP after consent authorisation	1-256

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ aspspId	ID of a financial institution servicing the Accounts of the PSU	ASPSP	Identification of ASPSP	M	1..1	UUID	Specifies the identification code of a financial institution which holds PSU accounts	36
++ merchant	Collect merchant data for reporting purposes	MC	Merchant data used for reporting or reconciliation purposes	O	0..1	Object	Set of elements used to define the merchant details	-
++++ id	Merchant identification code to identify the merchant	MC	Unique Merchant identifier per TPP, which could be used for reporting or reconciliation purposes	M	1..1	String	Field is available for TPPs to enable capturing of a merchant ID	1-256
++++ name	Merchant name to identify the merchant	MC	Name of merchant	M	1..1	String	Field is available for TPPs to enable capturing of a merchant name	1-256
paymentId	Unique identification as assigned by the ASPSP to uniquely identify the payment resource.	TPP	This value is used to make subsequent payment specific requests	M	1..1	String	PI / Transfer identification	1-256

Response – Success

HTTP Response Code = 200, OK

Note For a full list of response codes, including those returned in failure scenarios, see section Response Codes.

Response header

N/A

Response body

Table 84 Get Payment Status Request Response Body

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
originalRequestInfo	Returns original request information to the initiating party (TPP).	MC	This object contains original request information	M	1..1	Object	Original request information received from the initiating party	-
++ xRequestId	A memorable ID which could be used to support in a dispute	TPP	This element could be used for request-response tracking	M	1..1	UUID	Original xRequestId given by the client on request	36
payments	Specifies payments information in returned data	TPP	This element encapsulates all payments information	C	0..1	Object	Set of elements used to reference a payment instruction	-
++ transactionStatus	Specifies the status of the payment information group	TPP	Transaction Status of the payment resource	M	1.1	String	Status of the payment	One of the following ISO 20022 statuses: • ACSC • ACSP • PDNG • RCVD • RJCT • CANC

Name	Purpose	Required by	How it can be used	Condition	Multiplicity	Type	Description	Limitations / Parameters
++ statusReasonCode	Provides detailed information on the status reason.	TPP	Can only be used in case the status is equal to "RJCT"	O	0..1	String	Status Reason Code is used to add more information when a generic status code is returned by transaction Status field.	The codes for statusReasonCode are defined in ISO20022 - "ExternalStatusReason1Code". The following codes could be returned when RJCT status is raised: • AM04 • CURR • AM12 • DS24 • DS02 • AG08

For more information on errors received, see the error message structure described in Error Information.

9 Appendix

Inbound Service API for MC Connect hub – AIS & PIS YAML files



Note If you are unable to download these files, they are also available upon request by contacting the Open Banking development team or your Onboarding contact.

Proprietary Rights

The information contained in this document is proprietary and confidential to Mastercard International Incorporated, one or more of its affiliated entities (collectively "Mastercard"), or both.

This material may not be duplicated, published, or disclosed, in whole or in part, without the prior written permission of Mastercard.

Trademarks

Trademark notices and symbols used in this document reflect the registration status of Mastercard trademarks in the United States. Please consult with the Global Customer Service team or the Mastercard Law Department for the registration status of particular product, program, or service names outside the United States.

All third-party product and service names are trademarks or registered trademarks of their respective owners.

Disclaimer

Mastercard makes no representations or warranties of any kind, express or implied, with respect to the contents of this document. Without limitation, Mastercard specifically disclaims all representations and warranties with respect to this document and any intellectual property rights subsisting therein or any part thereof, including but not limited to any and all implied warranties of title, non-infringement, or suitability for any purpose (whether or not Mastercard has been advised, has reason to know, or is otherwise in fact aware of any information) or achievement of any particular result. Without limitation, Mastercard specifically disclaims all representations and warranties that any practice or implementation of this document will not infringe any third-party patents, copyrights, trade secrets or other rights.

Translation

A translation of any Mastercard manual, bulletin, release, or other Mastercard document into a language other than English is intended solely as a convenience to Mastercard customers. Mastercard provides any translated document to its customers "AS IS" and makes no representations or warranties of any kind with respect to the translated document, including, but not limited to, its accuracy or reliability. In no event shall Mastercard be liable for any damages resulting from reliance on any translated document. The English version of any Mastercard document will take precedence over any translated version in any legal proceeding.

Information Available Online

Mastercard provides details about the standards used for this document—including times expressed, language use, and contact information—on the Publications Support page available on Mastercard Connect. Go to Publications Support for centralized information.