EXECUTIVE BOARD
OF THE NATIONAL BANK OF MOLDOVA

DECISION

No 33 of 16 february 2026

on the approval of Functional and technical requirements for the interfaces of account servicing payment service providers

(in force as of 20.03.2026)

Official Monitor of the Republic of Moldova No 88-91, Article 137 of 20.02.2026

* * *

Pursuant to Article 52^4 paragraph (7) of Law No 114/2012 on payment services and electronic money (Official Monitor of the Republic of Moldova, 2012, No 193-197, Article 661), as amended, and point 65 of the Regulation on strong customer authentication and open, common, and secure standard of communication between payment service providers, approved by the Decision of the Executive Board No 12/2024 (published in the Official Monitor of the Republic of Moldova, 2024, No 36-39, Article 90), the Executive Board of the National Bank of Moldova.

DECIDES:

1. The Functional and technical requirements for the interfaces of account servicing payment service providers shall be approved (attached).
2. Payment service providers referred to in Article 5 paragraph (1) letters a)-c) of Law No 114/2012 on payment services and electronic money shall ensure compliance with requirements 4 and 5 of Table No 3, of the Functional and technical requirements for the interfaces of account servicing payment service providers, within 8 months from the date of entry into force of this decision.
3. This decision shall enter into force one month after its publication in the Official Monitor.

CHAIRMAN
OF THE EXECUTIVE BOARD
Anca-Dana DRAGU

Approved by

the Decision of the Executive Board of the National Bank of Moldova No 33/2026

Functional and technical requirements for the interfaces of account servicing payment service providers

CHAPTER I
GENERAL PROVISIONS

Section 1
Object and purpose

1. The Functional and technical requirements for the interfaces of account servicing payment service providers (hereinafter referred to as the *Requirements*) set out the functional, technical, and security conditions for the implementation of Open Banking by account servicing payment service providers (ASPSPs) in accordance with applicable national regulations and international standards.

2. The requirements represent the national standard for Open Banking and aim to ensure compliance with the provisions of Law 114/2012 on payment services and electronic money (hereinafter referred to *as Law No 114/2012*) and Regulation No 12/2024 on strong customer authentication and open, common, and secure standard of communication between payment service providers, approved by Decision No 12/2024 of the Executive Board of the National Bank of Moldova (hereinafter referred to as *Regulation No 12/2024*), facilitating interoperability with third-party payment service providers and creating a secure and standardized platform for users.

Section 2
Main notions

3. The terms and expressions used in these Requirements have the meaning provided in Law No 114/2012 and in the normative acts of the National Bank of Moldova (hereinafter - NBM), issued pursuant to Law No 114/2012.

4. Additionally, for the purposes of these Requirements, the following terms shall apply:

4.1. **Strong customer authentication** – authentication based on the use of two or more elements from the categories of knowledge (something only the user knows), possession (something only the user possesses), and inherence (something that represents the user). These elements are independent, and compromising one

element does not compromise the reliability of the other elements. These elements are also designed to protect the confidentiality of authentication data;

4.2. **The Berlin Group** – a group that defines technical standards for the implementation of Open Banking in accordance with Directive (EU) 2015/2366 of the European Parliament and of the Council of 25 November 2015 on payment services in the internal market, amending Directives 2002/65/EC, 2009/ 110/EC and 2013/36/EU and Regulation (EU) No 1093/2010 and repealing Directive 2007/64/EC (text with EEA relevance);

4.3. **Application Programming Interface** – represent the set of interfaces implemented by account servicing payment service providers, in accordance with the open, common, and secure standard of communication, which enable the secure and standardized exchange of information between participants, as well as the initiation of financial transactions on behalf of users, in compliance with the Requirements;

4.4. **Third-Party Provider** – authorized entity which, with the express consent of users, may access financial information in their accounts or initiate payments on their behalf;

4.5. **Redirect Method** – a method of authentication and authorization within Open Banking, whereby the user is redirected from the third-party provider's application to the account servicing payment service provider's application that offers account management services for the purpose of performing Strong Customer Authentication and authorizing a payment or data access;

4.6. **Dashboard** – represents a unified digital interface, made available to the user by the account servicing payment service provider, which allows the viewing, control, and management of all permissions granted to third-party providers to access the user's financial data;

4.7. **SCA** – Strong Customer Authentication;

4.8. **API** – Application Programming Interface;

4.9. **ASPSP** – Account Servicing Payment Service Providers;

4.10. **AISP** – Account Information Service Provider;

4.11. **PISP** – Payment Initiation Service Provider;

4.12. **TPP** – Third-Party Provider;

4.13. **AIS** – Accounting Information System;

4.14. **PIS** – Payment Initiation Services;

4.15. **Open Banking** – Open, common, and secure communication standard, pursuant to Article 52^4 paragraph (7) of Law No 114/2012 and point 3 of Regulation No 12/2024;

4.16. **PSU** – Payment Services User;

4.17. **URI** – Uniform Resource Identifier;

4.18. **CAS** – Central Addressing Scheme.

CHAPTER II
OPERATING MODEL AND TYPES OF SERVICES AVAILABLE IN OPEN BANKING

5. The Open Banking operating model allows secure and controlled access to users' financial information and the initiation of payments on their behalf, with the express consent of the account holder. Interaction between the parties involved is achieved through standardized APIs, which ensure interoperability, security, and protection of user data.

6. The following types of services are available in Open Banking:

6.1. AIS – this service allows TPPs to access and aggregate financial data from users' accounts with different payment service providers.

6.2. PIS – this service allows TPPs to initiate payments from users' accounts, with their consent, through the API provided by the ASPSP where they hold accounts. Users authorize transactions through the authentication mechanisms required by the respective provider, in accordance with SCA requirements.

7. ASPSP allows access to AIS and PIS services to TPPs that:

7.1. are registered to provide account information services or licensed to provide payment initiation services, in accordance with the provisions of Law No 114/2012;

7.2. uses public key certificates for authentication and electronic signature, in accordance with the provisions of Law No 124/2022 on electronic identification and trust services (hereinafter referred to as *Law No 124/2022*).

8. When implementing Open Banking, the ASPSP shall take into account the following functionalities:

8.1. Open Banking services are available to individuals;

8.2. access to Open Banking is provided through the mobile applications of payment service providers;

8.3. payment accounts are operated in the national currency (MDL), in accordance with Law No 114/2012;

8.4. payment products available in Open Banking include domestic A2A payments (domestic payments) and instant P2P payments (instant payments), offered in a single payment service ("payments") for initiating payments one at a time;

8.5. consent authorization and user authentication are performed using the redirect method.

CHAPTER III
FUNCTIONAL AND TECHNICAL REQUIREMENTS

Section 1
Functional and interface requirements

9. The implementation of Open Banking by ASPSPs must ensure an optimal experience for PSUs, guaranteeing security, accessibility, and transparency, and the requirements in Tables 1, 2, and 3 should focus on the intuitive interaction of PSUs with ASPSP applications, as well as compliance with security standards and applicable regulations in accordance with the provisions of Regulation No 12/2024.

10. ASPSP must ensure an intuitive design of the Open Banking interfaces in the mobile application, which is consistent and easy to navigate, allowing users to quickly find the information they need.

11. ASPSP must comply with the requirements for AIS, as set out in Table 1.

Table No 1

Requirement 1	ASPSP must apply SCA to access account data, excluding cases where an exception regulated in accordance with the provisions of Regulation No 12/2024 applies.
Requirement 2	The authentication process in the ASPSP mobile application must be equivalent to that which the PSU undergoes when directly accessing the mobile application (biometric, PIN code, credentials), without any additional steps.
Requirement 3	The ASPSP application must support app-to-app redirection, ensuring a secure and transparent transition between the AISP and the ASPSP.
Requirement 4	The ASPSP must display an intermediate screen indicating the status of the request and informing the PSU about the redirection to the AISP application. After confirmation of consent, the redirection must be performed automatically on the same device.
Requirement 5	The ASPSP must inform the PSU of the validity period and expiry date of the consent given to access the payment account data.
Requirement 6	ASPSP must allow the simultaneous sharing of multiple accounts held by PSU, without imposing limitations on their number.

Requirement 7	ASPSP must ensure that only active accounts are shared; blocked or closed accounts cannot be provided to TPP.
Requirement 8	In the event of closure or blocking of an account that is shared with a TPP on the basis of valid consent provided by the PSU, the ASPSP shall apply the following rules: Consent given by the PSU for a single account that has been closed or blocked: the consent is automatically revoked by the ASPSP, and upon request by the TPP, the ASPSP responds that access has been revoked and the resource (account) no longer exists. Consent given by the PSU for multiple accounts, where one of these accounts is closed or blocked: the consent remains valid for the remaining active accounts, and information will no longer be shared for the closed or blocked account. In the event the account is reopened or unblocked, a new consent will be required for that account.
Requirement 9	The ASPSP must provide the AISP with the requested data (e.g., balance, transactions) only within the limits of the valid consent given by the PSU
Requirement 10	The ASPSP must handle failure cases (unsuccessful authentication, lack of consent, technical errors, incorrect IBAN) and communicate the corresponding responses to the TPP.
Requirement 11	The ASPSP must ensure rapid processing (preferably within a few seconds) of authentication and authorisation requests, in order to maintain user trust and a smooth experience, particularly for payment initiation or transaction approval.
Requirement [TECH] 12	The ASPSP must limit the number of requests from the same AISP to a maximum of 4 calls within a 24-hour period, without any explicit action from the PSU.
Requirement [TECH] 13	The ASPSP must correctly handle the address (URI) to which the PSU will be redirected back to the AISP application after completing authentication or giving consent.

12. The ASPSP must comply with the requirements for PIS, as set out in Table 2.

Table No 2

Requirement 1	The ASPSP must apply SCA. By way of derogation, SCA shall not be applied in cases where the exceptions regulated by Regulation

	No 12/2024 occur.
Requirement 2	If the PSU has the ASPSP application installed on the same device, the redirection must invoke the ASPSP application only for authentication, without introducing additional screens.
Requirement 3	Authentication in the ASPSP application must involve no more steps than when the PSU accesses the application directly (biometric, PIN code, credentials, etc.).
Requirement 4	After successful authentication, the PSU must be redirected directly to the payment details page (amount, merchant, fee if applicable) to confirm or reject the payment.
Requirement 5	If the payment request does not include the debtor account and/or other data necessary to execute the payment, the ASPSP must provide the PSU with the option to select the account from which the payment will be made and/or to complete the missing information.
Requirement 6	The ASPSP must validate all payment data provided by the PISP and correctly handle error situations (incorrect IBAN, invalid data, blocked account, etc.), providing the appropriate response to the PISP.
Requirement 7	The ASPSP must display an intermediate screen indicating the status of the request and informing the PSU that they will be automatically redirected back to the PISP, without any further action required.
Requirement 8	Payment confirmation to the PISP and the PSU must be immediate, including a unique reference ID and the current status of the payment (pending, accepted, rejected).
Requirement 9	Payments must be processed in accordance with the provisions of Law 114/2012 and the normative acts issued by the NBM under Law 114/2012, including compliance with cut-off times, currency conversion if necessary, and reporting to the competent authorities.
Requirement 10	The ASPSP must allow real-time tracking of the payment status, accessible through both the PSU interface and the PISP interface. Notifications of successful or failed payment processing are recommended for transparency.
Requirement	In the case of an instant payment initiated by the PISP, the ASPSP must request the default account and other details from the CAS,

11	based on the alias provided in the creditorAccount field.
Requirement 12 [TECH]	All relevant actions (authentication, payment validation, confirmation/rejection) must be logged for full traceability and audit purposes.

13. The ASPSP must comply with the requirements and recommendations for PIS and AIS (included in a Dashboard), as set out in Table No 3.

Table No 3

Requirement 1	The ASPSP must provide the PSU with a centralized dashboard, accessible via the channels provided by the ASPSP (web and/or mobile application), where all active consents are displayed. For each TPP, the dashboard must show: the name of the TPP, the accounts concerned, the type of access granted (e.g., balance, transactions), the declared purpose of access, and the validity period of the consent.
Requirement 2	The information displayed on the dashboard must be updated in real time. Any action to revoke or modify a consent must be reflected immediately.
Requirement 3	The PSU must be able to revoke access granted to a TPP through a simple and intuitive action, and the revocation must take effect immediately.
Requirement 4	After revocation, the ASPSP must provide the PSU with a clear confirmation, including the name of the TPP, and the date and time when the revocation was carried out.
Requirement 5	The PSU must be proactively notified when a consent is approaching its expiry, so that they can decide whether to renew it or allow it to expire.
Recommendation 6	It is recommended that the dashboard also include a detailed history of TPP access, indicating the date of granting, modification, or revocation of consent, as well as the types of data accessed by the TPP during the relevant period.

Section 2
Technical requirements

14. The ASPSP must provide a technical infrastructure capable of ensuring interoperability with authorised TPPs, in compliance with these Requirements.

The implemented systems must guarantee the protection of PSU data, the integrity of transactions, and the continuity of the services provided.

15. The ASPSP must comply with the technical requirements set out in Table No 4.

Table No 4

Requirement 1	Compliance with API standards – within Open Banking in Moldova, the ASPSP is obliged to develop and maintain APIs in accordance with Annex No 1 to these Requirements. These APIs must be updated in a timely manner to ensure compatibility and ongoing compliance with the established requirements.
Requirement 2	Authentication and identity management – The ASPSP must implement mechanisms for the authentication and identification of TPPs in accordance with the provisions of Law No 124/2022, including the use of public key certificates where applicable, as well as procedures for signing and encrypting transmitted data.
Requirement 3	Data protection and privacy – The ASPSP is responsible for implementing advanced security measures, strict access control policies, and mechanisms to prevent leaks or unauthorized use of sensitive information.
Requirement 4	Incident management and reporting – The ASPSP must have an effective system for detecting, managing, and reporting security incidents, notifying the competent authorities and, where applicable, the affected PSUs. Depending on the nature of the incident, the ASPSP shall use the forms and methods of notification provided by the applicable normative framework.
Requirement 5	SCA – The ASPSP is responsible for applying SCA in all relevant flows, except in cases provided for in accordance with the provisions of Regulation No 12/2024. The implemented mechanisms must ensure a high level of security and fraud protection, using authentication methods equivalent to those available in the ASPSP's own channels (e.g., biometric, access code, password).
Requirement 6	Processing capacity and scalability – The ASPSP's technical infrastructure must be sized to efficiently handle large volumes of simultaneous requests from TPPs, maintaining adequate response times and continuous service availability.
Requirement 7	Testing, auditing, and certification of the IT systems used for the implementation of Open Banking shall be carried out in accordance with the requirements for systems/services related to critical ICT, as set out in the Regulation on minimum

	requirements for managing information and communication technology risks, information security, and business continuity, approved by Decision No 29/2025 of the Executive Board of the National Bank of Moldova.

16. The ASPSP must comply with the applicable national regulations regarding the provision of payment services, integration within Open Banking, and the protection of personal data.

Section 3
Specific interfaces

17. The ASPSP is obliged to provide TPPs with specific interfaces, dedicated to secure and controlled access to PSU account data, as well as to payment initiation on behalf of the PSU, based on the PSU's explicit consent.

18. The API must comply with Annex No 1 to these Requirements, which has been developed based on the Berlin Group standard and adapted to the normative and operational framework applicable in the Republic of Moldova.

19. The details regarding the API methods to be exposed by the ASPSP, the data structure, and the parameters required for integration are set out in Annex No 1 to these Requirements; the types of responses and standardized error codes are provided in Annex No 2 to these Requirements, and the mechanism for the ASPSP to verify calls made by TPPs is provided in Annex No 3 to these Requirements.

Section 4
List of mandatory methods

20. The methods to be developed by the ASPSP for providing consents are set out in Table No 5.

Table No 5

Endpoint	Method	Condition	Description
consents	POST	Mandatory	This method allows a TPP to request a PSU's consent to access account information. The consent is created in the ASPSP's information system and specifies the accounts and permissions granted.

consents/{consentId}	GET	Mandatory	This method provides details about an existing consent, identified by the consentId.
	DELETE	Mandatory	This method allows a TPP to revoke an active consent based on the corresponding action initiated by the PSU, thereby removing the TPP's access to the PSU's account data.
consents/{consentId}/status	GET	Mandatory	This method returns the current status of a consent.

21. The methods to be developed by the ASPSP for AIS are set out in Table No 6.

Table No 6

Endpoint	**Method**	**Condition**	**Description**
accounts	GET	Mandatory	This method provides a list of all accounts for which the PSU has granted the TPP access. The list includes basic details such as the IBAN, currency, and account type.
accounts/{account-id}	GET	Mandatory	This method provides detailed information about a specific account, identified by account-id, such as the account type, IBAN, and currency.
accounts/{account-id}/balances	GET	Mandatory	This method provides information about the balance of a specific account, including the available balance, booked balance, and expected balance.
accounts/{account-id}/transactions	GET	Mandatory	This method provides a list of transactions for a specific account, with options to filter by time period or transaction type (e.g., booked, pending).

22. The methods to be developed by the ASPSP for PIS are set out in Table No 7.

Table No 7

Endpoints	Method	Condition	Description
payments/{payment-product}	POST	Mandatory	This method initiates a payment using the specified payment product (e.g., domestic payments or instant payments).
payments/{payment-product}/{paymentId}	GET	Mandatory	This method provides detailed information about a specific payment, identified by paymentId, including the payment status, amount, and beneficiary.
payments/{payment-product}/{paymentId}/status	GET	Mandatory	This method provides the current status of a specific payment, such as pending, accepted, or rejected.

CHAPTER IV
OPEN BANKING DIGITAL LIST

23. For the implementation of Open Banking, the NBM has developed and manages the Open Banking Digital List – an interoperable digital platform, accessible via API, which ensures the recording and management of participants in the Open Banking ecosystem.

24. The Open Banking Digital List is intended for both ASPSPs and TPPs; the API provides data about each participant, contributing to the transparency and security of the system.

25. Payment service providers involved in Open Banking have access to relevant information according to their role, as follows:

25.1. ASPSPs receive the list of TPPs licensed and registered by the NBM. This list includes details such as the license number, the serial number of the public key certificate (in hexadecimal format) used by the TPP, and other information necessary to validate their access.

25.2. TPPs receive the list of ASPSPs integrated into the Open Banking ecosystem, along with details regarding the API endpoints available for accessing data.

26. To ensure compliance with the requirement set out in point 7, ASPSPs may access the Open Banking Digital List in accordance with the technical connection details provided by the NBM. For real-time verification of information regarding a TPP

based on the Open Banking Digital List, ASPSPs shall query the Open Banking Digital List at intervals of between 1 and 10 minutes. Between two successive queries, ASPSPs may use the cached version of the list until the next update. If the Open Banking Digital List is unavailable, ASPSPs may use data from the most recently cached version.

27. ASPSPs must continuously monitor the status of the public key certificates associated with TPPs and refuse access to any certificate that is expired, compromised, or revoked.

28. If, following verification, the ASPSP detects discrepancies between the information it holds regarding a TPP and the information provided in the TPP's request, it is recommended that the ASPSP block access and notify the NBM via the email open.banking@bnm.md. Upon receiving the notification, the NBM will review the information in accordance with its internal procedures and inform the ASPSP of the results of the review within 5 business days.

29. If an ASPSP identifies the need for additional functionalities or technical modifications to existing ones, it shall submit an official request to the NBM. The NBM is entitled to reject such requests if, following analysis, they do not (cumulatively) meet the following criteria:

29.1. the request does not contravene the provisions of the normative acts;

29.2. the request is necessary and consistent with Open Banking functionalities.

Annex No 1

API methods to be exposed by the ASPSP, the data structure, and the parameters required for integration

1. Consents Endpoints

Method: Create Consent

Method type: POST /v1/consents

Description: This method allows the creation of an access consent for a PSU. A consent is required to authorise a TPP to access the account information (AIS).

[TPP] Request Header Parameters:

Name	Type	Condition	Description
X-Request-ID	UUID	Mandatory	Unique ID generated by the TPP to identify the request. Mandatory for verification of uniqueness by the ASPSP.
PSU-IP-Address	String	Mandatory	The IP address of the PSU.
PSU-Device-ID	String	Mandatory	The unique ID of the device used by the PSU.
PSU-Device-Name	String	Mandatory	Name/model (generic) of the device to which the end user connects (PSU).
PSU-Geo-Location	String	Optional	The geographic location transmitted with the corresponding HTTP request between the PSU and the TPP, if available.
TPP-Redirect-URI	String	Mandatory	The redirect URI to TPP after consent completion.
TPP-Nok-Redirect-URI	String	Optional	The redirect URI in case of error. ASPSP has the right to ignore this field.
Date	Datetime	Mandatory	The date and time when the request is made (RFC 7231 D, dd M YYYY HH:mm:ss GMT).
Content-Type	String	Mandatory	Specifies the request body format application/json.
Digest	String	Mandatory	It is included only if and only if the “Signature” element is included in the request header. It is recommended to calculate the digest after applying

Name	Type	Condition	Description
			JSON Minify (this recommendation applies throughout the document).
Signature	String	Mandatory	Signing of the request by the TPP at the application level. To complete the field, a “signing string” is generated according to the “algorithm” and “headers” parameters, signed with the private key associated with “keyId”, and the base64-encoded result is then inserted into the “signature” field. This method of completing the “Signature” field applies throughout this document.
TPP-Signature-Certificate	String	Mandatory	The certificate used to sign the request, in base64 encoding. It must be included if a signature is present.

[TPP] Request Header Sample:

POST htttps://api.provider.com/v1/consents

Content-Type: application/json

X-Request-ID: 123e4567-e89b-12d3-a456-426614174000

PSU-IP-Address: 192.168.0.10

PSU-Device-ID: device-12345

PSU-Device-Name: ModelDevice X

PSU-Geo-Location: GEO: 47.014434;28.493426

TPP-Redirect-URI: https://tpp-example.md/redirect

TPP-Nok-Redirect-URI: https://tpp-example.md/redirect-failure

Date: Wed, 11 Sep 2024 12:34:56 GMT

Digest: SHA-256=VYe+GLeBVnBVH8A50NP0Cawtg1xwkfe+XufPzmVGGMA=

Signature: keyId=“SN= 4000000010FC01D520258AB15EAF, CA=CN=D-eSystemTrustIB, O=IP STISC 1003600096694, C=MD”, algorithm=”rsa-sha256”,

headers=”digest date x-request-id tpp-redirect-uri”,

signature=”Base64(RSA-SHA256(signing string))”

TPP-Signature-Certificate: "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzKzT+I32ygAqDdZVfKYtDkWVZT7ySP54ZXgH8dEUM6d9fKhs6DFiM9Do5slDDo7YwLjXU8Iq7C4eONHp+7u0z5LmvMyYnxgD0h1S7

F6T5gqaOQz3Qkm9bW2QY5M6Fh8/FivYpno3pzUNrzzTyAdIQ8MjbbJff7cDwDpwnFVgbQ6ZTx
Ym2CccovJQJuyfwO7ICtVjkkXq+FXWmZTfl2AfQwvMFuPRTlxjDLDBMOwDsYMBVBym8vSdzY
7AkDPylQtD/kTxM+oLlo7mMtpTeDs/qhvZXMnRPvE/JIE58xsiCBvUe36V1ht+WLidqk9iYxeAwT
bF7kZgxXjUGBYDrz/B4fqa2FqNzdsq2+LfsAk5cDBshXq1t/vmhty7TK09KPBrbDAjm9uDbf6zA0Z
SczX4rh7tBf3rc5BC+MBuLKgg1Pv9WgfWHi5BQ==",

Where „*signing string*" is:

digest: SHA-256=VYe+GLeBVnBVH8A50NP0Cawtg1xwkfe+XufPzmVGGMA=
date: Wed, 11 Sep 2024 12:34:56 GMT
x-request-id: 123e4567-e89b-12d3-a456-426614174000
tpp-redirect-uri: https://tpp-example.md/redirect

[TPP] Request Body Sample for Consent Request on Dedicated Accounts (detailed consent – support by ASPSP is mandatory):

```
{
 "access": {
  "accounts": [
   {
    "iban": "MD21AAA000000022553456789"
   }
  ],
  "balances": [
   {
    "iban": "MD21AAA000000022553456789"
   }
  ],
  "transactions": [
   {
    "iban": "MD21AAA000000022553456789"
   }
  ]
 },
 "recurringIndicator": true,
 "validUntil": "2024-12-31",
 "frequencyPerDay": 4
```

```
}
```

[TPP] Request Body Sample for Consent on Account List of Available Accounts (global consent - support by ASPSP is mandatory):

```
{
 "access": {
        "availableAccounts": "allAccounts"
},
        "recurringIndicator": true,
        "validUntil": "2025-08-06",
        "frequencyPerDay": 4
}
```

[TPP] Request Body Sample for Consent on Account List or without Indication of dedicated Accounts (bank offered consent - support by ASPSP is mandatory):

```
{
  "access": {
    "balances": [],
    "transactions": []
  },
  "recurringIndicator": true,
  "validUntil": "2025-11-01",
  "frequencyPerDay": 4
}
```

[TPP] Request Body Parameters:

Parameter	Type	Condition	Description
access	Object	Mandatory	Defines access to accounts, balances, and transactions.
access.accounts	Array	Optional	List of IBANs of the accounts to which the TPP will have access.
access.balances	Array	Optional	List of IBANs for which the TPP may access the balances.

Parameter	Type	Condition	Description
access.transactions	Array	Optional	List of IBANs for which the TPP may access transaction details.
recurringIndicator	Boolean	Mandatory	Indicates whether access is recurrent or one-time. If true, access is recurrent. If false, access is one-time.
validUntil	String	Mandatory	Consent expiry date (format: YYYY-MM-DD).
frequencyPerDay	Integer	Mandatory	The maximum number of accesses per day to account data without PSU involvement (AIS). Possible values are 1, 2, 3, or 4.

[ASPSP] Response Header sample:

```
Content-Type: application/json
X-Request-ID: 123e4567-e89b-12d3-a456-426614174000
ASPSP-SCA-Approach: REDIRECT
Date: Wed, 11 Sep 2024 12:34:58 GMT
Location: “/v1/consents/ a7c3e9f8-1a44-4cd3-83ab-4f29d1f9e8c7”
```

[ASPSP] Response Body Sample (Success - 201 Created):

```
{
 "consentStatus": "received",
 "consentId": "d6f9b8f4-4b10-4b9e-933b-ff9a24b5641f",
 "_links": {
  "scaRedirect": {
   "href": "www.mybankapp.com//authentication/ a7c3e9f8-1a44-4cd3-83ab-4f29d1f9e8c7"
  },
  "status": {
   "href": "/v1/consents/ a7c3e9f8-1a44-4cd3-83ab-4f29d1f9e8c7 /status"
  },
  "scaStatus": {
   "href": "/v1/consents/ a7c3e9f8-1a44-4cd3-83ab-4f29d1f9e8c7 /authorisations/123auth567"
  }
 }
```

```
}
```

[ASPSP] Response Header Parameters:

Name	Type	Condition	Description
X-Request-ID	UUID	Mandatory	Unique ID generated by TPP to identify the request.
Location	String	Mandatory	Location of the created resource.
ASPSP-SCA-Approach	String	Mandatory	The type of SCA method used by ASPSP. Currently only REDIRECT.
Date	Datetime	Mandatory	The date and time when the request is made (RFC 7231).
Content-Type	String	Mandatory	Specifies that the response body format is application/json.

[ASPSP] Parameters of Success Response:

Parameter	Type	Condition	Description
consentStatus	String	Mandatory	Consent status (valid, expired, etc.).
consentId	String	Mandatory	The unique ID of the created consent.
_links	String	Mandatory	List of links associated with a consent.

[ASPSP] Response Sample (Error - 400 Bad Request):

```
{
  "tppMessages": [
    {
      "category": "ERROR",
      "code": "FORMAT_ERROR",
      "text": "string",
      "path": "string"
    }
  ]
}
```

[ASPSP] Parameters of Error Response:

Parameter	Type	Condition	Description
tppMessages	Array	Mandatory	List of error messages generated by the server.
tppMessages.category	String	Mandatory	Message category (ERROR, WARNING).
tppMessages.code	String	Mandatory	Error code (e.g., FORMAT_ERROR)
tppMessages.text	String	Mandatory	Detailed description of the error (e.g., "Invalid IBAN format").
tppMessages.path	String	Optional	Indicates the exact location of the error.

Method: Get Consent

Method type: GET /v1/consents/{consentId}

Description: This method allows the TPP to obtain detailed information about a previously created consent. The consent ID (consentId) is provided in the URL. The TPP can check the status of the consent and associated details, such as the accounts and validity period.

[TPP] Path Parameters:

Name	Type	Condition	Description
consentId	String	Mandatory	The unique ID of the created consent.

[TPP] Request Header Parameters:

Name	Type	Condition	Description
X-Request-ID	UUID	Mandatory	Unique ID generated by the TPP to identify the request.
Date	Datetime	Mandatory	The date and time when the request is made (RFC 7231).
PSU-IP-Adress	String	Mandatory	The IP address of the PSU. In the case of a call without PSU involvement, the TPP fills in 0.0.0.0.
PSU-Device-ID	String	Mandatory	The unique ID of the device used by the PSU. In the case of a call without PSU involvement, the TPP fills in no-psu-involved.
PSU-Device-Name	String	Mandatory	The name/model (generic) of the device from which the PSU connects. In the case of a call without PSU involvement, the TPP fills in no-psu-involved.

Name	Type	Condition	Description
PSU-Geo-Location	String	Optional	The transmitted geographical location of the HTTP request between the PSU and TPP, if available.
Digest	String	Mandatory	It is included only if and only if the “Signature” element is included in the request header.
Signature	String	Mandatory	Application-level signing of the request by the TPP.
TPP-Signature-Certificate	String	Mandatory	The certificate used to sign the request, in base64 encoding. It must be included if a signature is present.

[TPP] Request Header Sample:

GET https://api.provider.com/v1/consents/ d6f9b8f4-4b10-4b9e-933b-ff9a24b5641f

X-Request-ID: 123e4567-e89b-12d3-a456-426614174000

Date: Wed, 11 Sep 2024 12:34:56 GMT

PSU-IP-Address: 192.168.0.10

PSU-Device-ID: device-12345

PSU-Device-Name: ModelDevice X

PSU-Geo-Location: GEO: 47.014434;28.493426

Digest: SHA-256=47DEQpj8HBSa+/TImW+5JCeuQeRkm5NMpJWZG3hSuFU=

Signature: keyId=“SN= 4000000010FC01D520258AB15EAF, CA=CN=D-eSystemTrustIB, O=IP STISC 1003600096694, C=MD”, algorithm=”rsa-sha256”,

headers=”digest date x-request-id”,

signature=”Base64(RSA-SHA256(signing string))”

TPP-Signature-Certificate:
"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzKzT+I32ygAqDdZVfKYtDkWVZT7y
SP54ZXgH8dEUM6d9fKhs6DFiM9Do5slDDo7YwLjXU8Iq7C4eONHp+7u0z5LmvMyYnxgD0h1S7
F6T5gqaOQz3Qkm9bW2QY5M6Fh8/FivYpno3pzUNrzzTyAdIQ8MjbbJff7cDwDpwnFVgbQ6ZTx
Ym2CccovJQJuyfwO7ICtVjkkXq+FXWmZTfl2AfQwvMFuPRTlxjDLDBMOwDsYMBVBym8vSdzY
7AkDPylQtD/kTxMo+4t7mMtpTeDs/qhvZXMnRPvE/JIE58xsiCBvUe36V1ht+WLidqk9iYxeAwTb
F7kZgxXjUGBYDrz/B4fqa2FqNzdsq2+LfsAk5cDBshXq1t/vmhty7TK09KPBrbDAjm9uDbf6zA0ZS
czX4rh7tBf3rc5BC+MBuLKgg1Pv9WgfWHi5BQ==",

[ASPSP] Response Header sample:

X-Request-ID: 123e4567-e89b-12d3-a456-426614174000

[ASPSP] Response Body Sample (Success - 200 OK):

```
{
  "access": {
    "accounts": [
      {
        "iban": "MD21AAA000000022553456789",
        "currency": "MDL"
      }
    ],
    "balances": [
      {
        "iban": "MD21AAA000000022553456789",
        "currency": "MDL"
      }
    ],
    "transactions": [
      {
        "iban": "MD21AAA000000022553456789",
        "currency": "MDL"
      }
    ]
  },
  "recurringIndicator": true,
  "validUntil": "2024-12-31",
  "frequencyPerDay": 4,
  "consentStatus": "valid",
"_links": {
"account": {
"href": "/v1/accounts"
    }
  }
```

```
}
```

[ASPSP] Response Header Parameters:

Name	Type	Condition	Description
X-Request-ID	UUID	Mandatory	Unique ID generated by the TPP to identify the request.

[ASPSP] Parameters of Success Response:

Parameter	Type	Condition	Description
consentStatus	String	Mandatory	Consent status (valid, expired, etc.).
access	Object	Mandatory	Defines access to accounts, balances, and transactions.
access.accounts	Array	Optional	List of IBANs of the accounts to which the TPP has access.
access.balances	Array	Optional	List of IBANs for which the TPP may access balances.
access.transactions	Array	Optional	List of IBANs for which the TPP may access transaction details.
currency	String	Optional	Account currency.
validUntil	String	Mandatory	Consent expiry date.
recurringIndicator	Boolean	Mandatory	Indicates whether access is recurring or one-time.
frequencyPerDay	Integer	Mandatory	Maximum number of accesses per day.
_links	Object	Optional	Indicates the next steps that can be taken through the interface

[ASPSP] Response Sample (Error - 400 Bad Request):

```
{
  "tppMessages": [
    {
      "category": "ERROR",
      "code": "FORMAT_ERROR",
      "text": "string"
      "path": "string"
```

```
  }
 ]
}
```

Parameter	Type	Condition	Description
tppMessages	Array	Mandatory	List of error messages generated by the server.
tppMessages.category	String	Mandatory	Message category (ERROR, WARNING).
tppMessages.code	String	Mandatory	Error code (e.g., FORMAT_ERROR).
tppMessages.text	String	Mandatory	Detailed description of the error (e.g., "Invalid IBAN format").
tppMessages.path	String	Optional	Indicates the exact location of the error.

Method: Delete Consent

Method type: DELETE /v1/consents/ {consentId}

Description: This method allows the TPP to revoke an existing consent using the consentId. Once revoked, the consent will no longer allow access to the PSU's accounts. This action is necessary to stop/block access to account data or to interrupt recurring access.

[TPP] Path Parameters:

Name	Type	Condition	Description
consentId	String	Mandatory	The unique ID of the created consent.

[TPP] Request Header Parameters:

Name	Type	Condition	Description
X-Request-ID	UUID	Mandatory	Unique ID generated by the TPP to identify the request. Mandatory for uniqueness verification by the ASPSP.
PSU-IP-Adress	String	Mandatory	The IP address of the PSU.
PSU-Device-ID	String	Mandatory	The unique ID of the device used by the PSU.
PSU-Device-Name	String	Mandatory	The name/model (generic) of the device from which the PSU connects.

Name	Type	Condition	Description
PSU-Geo-Location	String	Optional	The transmitted geographical location of the corresponding HTTP request between the PSU and TPP, if available.
Date	String	Mandatory	The date and time when the request is made. (RFC 7231)
Digest	String	Mandatory	It is included only if and only if the “Signature” element is included in the request header.
Signature	String	Mandatory	Application-level signing of the request by the TPP.
TPP-Signature-Certificate	String	Mandatory	The certificate used to sign the request, in base64 encoding. It must be included if a signature is present.

[TPP] Request Header Sample:

DELETE htttps://api.provider.com/v1/consents/ d6f9b8f4-4b10-4b9e-933b-ff9a24b5641f

X-Request-ID: 123e4567-e89b-12d3-a456-426614174000

PSU-IP-Address: 192.168.0.10

PSU-Device-ID: device-12345

PSU-Device-Name: ModelDevice X

PSU-Geo-Location: GEO: 47.014434;28.493426 Date: Wed, 11 Sep 2024 12:34:56 GMT

Digest: SHA-256=47DEQpj8HBSa+/TImW+5JCeuQeRkm5NMpJWZG3hSuFU=

Signature: keyId=“SN= 4000000010FC01D520258AB15EAF, CA=CN=D-eSystemTrustIB, O=IP STISC 1003600096694, C=MD”, algorithm=”rsa-sha256”,

headers=”digest date x-request-id”,

signature=”Base64(RSA-SHA256(signing string))”

TPP-Signature-Certificate:
"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzKzT+I32ygAqDdZVfKYtDkWVZT7y
SP54ZXgH8dEUM6d9fKhs6DFiM9Do5slDDo7YwLjXU8Iq7C4eONHp+7u0z5LmvMyYnxgD0h1S7
F6T5gqaOQz3Qkm9bW2QY5M6Fh8/FivYpno3pzUNrzzTyAdIQ8MjbbJff7cDwDpwnFVgbQ6ZTx
Ym2CccovJQJuyfwO7ICtVjkkXZTfl2AfQwvMFuPRTlxjDLDBMOwDsYMBVBym8vSdzY7AkDPyl
QtD/kTxMo+4toD+oLlo7mMtpTeDs/qhvZXMnRPvE/JIE58xsiCBvUe36V1ht+WLidqk9iYxeAwTb
F7kZgxXjUGBYDrz/B4fqa2FqNzdsq2+LfsAk5cDBshXq1t/vmhty7TK09KPBrbDAjm9uDbf6zA0ZS
czX4rh7tBf3rc5BC+MBuLKgg1Pv9WgfWHi5BQ==”

[ASPSP] Response Header sample:

X-Request-ID: 123e4567-e89b-12d3-a456-426614174000

[ASPSP] Response Body Sample (Success - 204 No Content):

No information is returned in the response body, only the 204 No Content status is provided, which confirms that the revocation was successful.

[ASPSP] Response Header Parameters:

Name	Type	Condition	Description
X-Request-ID	UUID	Mandatory	Unique ID generated by the TPP to identify the request.

[ASPSP] Response Sample (Error - 400 Bad Request):

```
{
 "tppMessages": [
  {
   "category": "ERROR",
   "code": "FORMAT_ERROR",
   "text": "string",
   "path": "string"
  }
 ]
}
```

Parameter	Type	Condition	Description
tppMessages	Array	Mandatory	List of error messages generated by the server.
tppMessages.category	String	Mandatory	Message category (ERROR, WARNING)
tppMessages.code	String	Mandatory	Error code (e.g., FORMAT_ERROR).
tppMessages.text	String	Mandatory	Detailed description of the error (e.g., “Invalid IBAN format”).
tppMessages.path	String	Optional	Indicates the exact location of the error.

Method: Consent status

Method type: GET /v1/consents/{consentId}/status

Description: This method allows the TPP to obtain the current status of a consent identified by the consentId. The consent status indicates whether it is valid, expired, revoked, rejected, or in the initial—received—stage, providing important information about the state of the granted consent.

[TPP] Path Parameters:

Name	Type	Condition	Description
consentId	String	Mandatory	The unique ID of the created consent.

[TPP] Request Header Parameters:

Name	Type	Condition	Description
X-Request-ID	UUID	Mandatory	Unique ID generated by the TPP to identify the request.
PSU-IP-Adress	String	Mandatory	The IP address of the PSU. In the case of a call without PSU involvement, the TPP fills in 0.0.0.0.
PSU-Device-ID	String	Mandatory	The unique ID of the device used by the PSU. In the case of a call without PSU involvement, the TPP fills in no-psu-involved.
PSU-Device-Name	String	Mandatory	The name/model (generic) of the device from which the PSU connects. In the case of a call without PSU involvement, the TPP fills in no-psu-involved.
PSU-Geo-Location	String	Optional	The transmitted geographical location of the corresponding HTTP request between the PSU and TPP, if available.
Date	String	Mandatory	The date and time when the request is made (RFC 7231).
Digest	String	Mandatory	It is included only if and only if the “Signature” element is included in the request header.
Signature	String	Mandatory	Application-level signing of the request by the TPP.
TPP-Signature-Certificate	String	Mandatory	The certificate used to sign the request, in base64 encoding. It must be included if a signature is present.

[TPP] Request Header Sample:

```
GET https://api.provider.com/v1/consents/ d6f9b8f4-4b10-4b9e-933b-ff9a24b5641f/status

X-Request-ID: 123e4567-e89b-12d3-a456-426614174000

PSU-IP-Address: 192.168.0.10

PSU-Device-ID: device-12345

PSU-Device-Name: ModelDevice X
```

PSU-Geo-Location: GEO: 47.014434;28.493426

Date: Wed, 11 Sep 2024 12:34:56 GMT

Digest: SHA-256=47DEQpj8HBSa+/TImW+5JCeuQeRkm5NMpJWZG3hSuFU=

Signature: keyId="SN= 4000000010FC01D520258AB15EAF, CA=CN=D-eSystemTrustIB, O=IP STISC 1003600096694, C-MD", algorithm="rsa-sha256",

headers="digest date x-request-id",

signature="Base64(RSA-SHA256(signing string))"

TPP-Signature-Certificate:
"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzKzT+I32ygAqDdZVfKYtDkWVZT7y SP54ZXgH8dEUM6d9fKhs6DFiM9Do5slDDo7YwLjXU8Iq7C4eONHp+7u0z5LmvMyYnxgD0h1S7 F6T5gqaOQz3Qkm9bW2QY5M6Fh8/o3pzUNrzzTyAdIQ8MjbbJff7cDwDpwnFVgbQ6ZTxYm2Ccc ovJQJuyfwO7ICtVjkkXq+FXWmZTfl2AfQwvMFuPRTlxjDLDBMOwDsYMBVBym8vSdzY7AkDPy lQtD/kTxMo+4toD+oLlo7mMtpTeDs/qhvZXMnRPvE/JIE58xsiCBvUe36V1ht+WLidqk9iYxeAwT bF7kZgxXjUGBYDrz/B4fqa2FqNzdsq2+LfsAk5cDBshXq1t/vmhty7TK09KPBrbDAjm9uDbf6zA0Z SczX4rh7tBf3rc5BC+MBuLKgg1Pv9WgfWHi5BQ=="

[ASPSP] Response Header sample:

X-Request-ID: 123e4567-e89b-12d3-a456-426614174000

[ASPSP] Response Body Sample (Success - 200 OK):


```
{
 "consentStatus": "valid"
}
```


[ASPSP] Response Header Parameters:

Name	Type	Condition	Description
X-Request-ID	UUID	Mandatory	Unique ID generated by the TPP to identify the request.

[ASPSP] Parameters of Success Response:

Parameter	Type	Condition	Description
consentStatus	String	Mandatory	The current status of the consent (valid, expired, revoked, rejected, etc.).

[ASPSP] Response Sample (Error - 400 Bad Request):


```
{
 "tppMessages": [
  {
```

```
    "category": "ERROR",
    "code": "FORMAT_ERROR",
    "text": "string",
    "path": "string"
   }
 ]
}
```


[ASPSP] Parameters of Error Response:

Parameter	Type	Condition	Description
tppMessages	Array	Mandatory	List of error messages generated by the server.
tppMessages.category	String	Mandatory	Message category (ERROR, WARNING).
tppMessages.code	String	Mandatory	Error code (e.g., FORMAT_ERROR).
tppMessages.text	String	Mandatory	Detailed description of the error (e.g., "Invalid IBAN format").
tppMessages.path	String	Optional	Indicates the exact location of the error.

2. Accounts Endpoint

Method: Read account list

Method type: GET /v1/accounts

Description: This method allows the TPP to obtain a list of accounts held by the PSU for which there is a valid consent. The returned data includes essential information about each account, such as the IBAN, currency, and other account details. This method is essential in the context of Account Information Services (AIS).

[TPP] Query Parameters:

Name	Type	Condition	Description
withBalance	Boolean	Optional	If included, this parameter must have the value true or false. If the parameter is true, the function will return the list of accessible payment accounts, including the account balance, to the extent that this has been granted by the PSU through consent and is available from

Name	Type	Condition	Description
			the ASPSP. If the parameter is false or not provided, the list of accounts will be returned without balances.

[TPP] Request Header Parameters:

Name	Type	Condition	Description
X-Request-ID	UUID	Mandatory	Unique ID generated by the TPP to identify the request.
Consent-ID	String	Mandatory	Unique ID of the consent based on which the TPP has access to the accounts.
PSU-IP-Address	String	Mandatory	The IP address of the PSU. In the case of a call without PSU involvement, the TPP fills in 0.0.0.0.
PSU-Device-ID	String	Mandatory	The unique ID of the device used by the PSU. In the case of a call without PSU involvement, the TPP fills in no-psu-involved.
PSU-Device-Name	String	Mandatory	The name/model (generic) of the device from which the PSU connects. In the case of a call without PSU involvement, the TPP fills in no-psu-involved.
PSU-Geo-Location	String	Optional	The transmitted geographical location of the corresponding HTTP request between the PSU and TPP, if available.
Date	Datetime	Mandatory	The date and time when the request is made (RFC 7231).
Digest	String	Mandatory	It is included if and only if the "Signature" element is included in the request header.
Signature	String	Mandatory	Application-level signing of the request by the TPP.
TPP-Signature-Certificate	String	Mandatory	The certificate used to sign the request, in base64 encoding. It must be included if a signature is present.

[TPP] Request Header Sample:

```
GET https://api.provider.com/v1/accounts

X-Request-ID: 123e4567-e89b-12d3-a456-426614174000

Consent-ID: d6f9b8f4-4b10-4b9e-933b-ff9a24b5641f
```

PSU-IP-Address: 192.168.0.10

PSU-Device-ID: device-12345

PSU-Device-Name: ModelDevice X

Date: Wed, 11 Sep 2024 12:34:56 GMT

Digest: SHA-256=47DEQpj8HBSa+/TImW+5JCeuQeRkm5NMpJWZG3hSuFU=

Signature: keyId="SN= 4000000010FC01D520258AB15EAF, CA=CN=D-eSystemTrustIB, O=IP STISC 1003600096694, C-MD", algorithm="rsa-sha256",

headers="digest date x-request-id",

signature="Base64(RSA-SHA256(signing string))"

TPP-Signature-Certificate:
"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzKzT+I32ygAqDdZVfKYtDkWVZT7y
SP54ZXgH8dEUM6d9fKhs6DFiM9Do5slDDo7YwLjXU8Iq7C4eONHp+7u0z5LmvMyYnxgD0h1S7
F6T5gqaOQz3Qkm9bW2QY5M6Fh8/FivYpno3pzUNrzzTyAdIQ8MjbbJff7cDwDpwnFVgbQ6ZTx
Ym2CccovJQJuyfwO7ICtVjkkXZTfl2AfQwvMFuPRTlxjDLDBMOwDsYMBVBym8vSdzY7AkDPyl
QtD/kTxMo+4toD+oLlo7mMtpTeDs/qhvZXMnRPvE/JIE58xsiCBvUe36V1ht+WLidqk9iYxeAwTb
F7kZgxXjUGBYDrz/B4fqa2FqNzdsq2+LfsAk5cDBshXq1t/vmhty7TK09KPBrbDAjm9uDbf6zA0ZS
czX4rh7tBf3rc5BC+MBuLKgg1Pv9WgfWHi5BQ=="

[ASPSP] Response Header sample:

X-Request-ID: 123e4567-e89b-12d3-a456-426614174000

[ASPSP] Response Body Sample (Success - 200 OK):


```
{
 "accounts": [
  {
   "resourceId": "acc-123456789",
   "iban": "MD21AAA000000022553456789",
   "currency": "MDL",
   "product": "Cont Curent",
   "cashAccountType": "CACC",
   "_links": {
    "balances": {
     "href": "/v1/accounts/ acc-123456789/balances"
    },
    "transactions": {
```

```
        "href": "/v1/accounts/ acc-123456789/transactions"
      }
    }
  },
  {
    "resourceId": "acc-987654321",
    "iban": "MD21BBB000000022553456999",
    "currency": "MDL",
    "product": "Cont de Economii",
    "cashAccountType": "SVGS",
    "_links": {
      "balances": {
        "href": "/v1/accounts/ acc-987654321/balances"
      }
    }
  }
 ]
}
```


[ASPSP] Response Body Sample withBalance (Success - 200 OK):


```
{
  "accounts": [
    {
      "resourceId": "acc-123456789",
      "iban": "MD21AAA000000022553456789",
      "currency": "MDL",
      "product": "Cont MDL",
      "cashAccountType": "CACC",
      "balances": [
        {
          "balanceType": "interimAvailable",
```

```
            "balanceAmount": {
               "currency": "MDL",
               "amount": "99999.99"
            },
            "lastChangeDateTime": "2024-08-25T00:00:00Z"
         }
      ],
      "_links": {
         "transactions": {
            "href": "/v1/accounts/acc-123456789/transactions"
         }
      }
   }
]
}
```


[ASPSP] Response Header Parameters:

Name	Type	Condition	Description
X-Request-ID	UUID	Mandatory	Unique ID generated by TPP to identify the request.

[ASPSP] Parameters of Success Response:

Parameter	Type	Condition	Description
accounts	Array	Mandatory	The list of accounts held by the PSU.
resourceId	String	Mandatory	The unique ID associated with each account.
iban	String	Mandatory	The account's IBAN.
currency	String	Mandatory	The account's currency (e.g., MDL).
product	String	Mandatory	The type of product associated with the account (e.g., Current Account).
cashAccountType	String	Mandatory	The account type (e.g., CACC for current account, SVGS for savings account).

Parameter	Type	Condition	Description
balances	Array	Mandatory (if withBalance)	The list of balances associated with the account.
balanceType	String	Mandatory (if withBalance)	The balance type (e.g., interimAvailable, expected).
balanceAmount	Object	Mandatory (if withBalance)	Details about the account balance.
balanceAmount.currency	String	Mandatory (if withBalance)	The account's currency (e.g., MDL).
balanceAmount.amount	String	Mandatory (if withBalance)	Amount available in account.
lastChangeDateTime	String	Mandatory (if withBalance)	Date and time of the last balance change.
_links	Object	Optional	Indicates the next steps that can be followed through the interface.

[ASPSP] **Response Sample (Error - 400 Bad Request):**


```
{
  "tppMessages": [
    {
      "category": "ERROR",
      "code": "CONSENT_INVALID",
      "text": "string"
      "path": "string"
    }
  ]
}
```


[ASPSP] **Parameters of Error Response:**

Parameter	Type	Condition	Description
tppMessages	Array	Mandatory	List of error messages generated by the server.
tppMessages.category	String	Mandatory	Message category (ERROR, WARNING).
tppMessages.code	String	Mandatory	Error code (e.g., FORMAT_ERROR).
tppMessages.text	String	Mandatory	Detailed description of the error (e.g., "Invalid IBAN format").
tppMessages.path	String	Optional	Indicates the exact location of the error.

Parameters cashAccountType (examples according to ISO 20022):

Code	Description
CACC	Current Account. This is used for daily transactions.
SVGS	Savings Account. This is used for saving money.
CASH	Cash Account. Used to reflect cash accounts that are not explicitly "current" or "savings" accounts.
TDEP	Term Deposit Account. Used for fixed-term deposit accounts.
LOAN	Loan Account. Accounts used for managing loans.
SLRY	Accounts used for salary payments.

Method: Read account details

Method type: GET /v1/accounts/{account-id}

Description: This method allows the TPP to obtain details about a specific account of the PSU, identified by the **accountId**. The returned data includes essential information about the account, such as the IBAN, currency, and other relevant details. This method is essential for providing Account Information Services (AIS).

[TPP] Path Parameters:

Name	Type	Condition	Description
account-id	String	Mandatory	The unique account ID assigned through the resourceId.

[TPP] Query Parameters:

Name	Type	Condition	Description
withBalance	Boolean	Optional	If included, this parameter must have the value true or false. If the parameter is true, the function will return the list of accessible payment accounts, including the account balance, to the extent that this has been granted by the PSU through consent and is available from the ASPSP. If the parameter is false or not provided, the list of accounts will be returned without balances.

[TPP] Request Header Parameters:

Name	Type	Condition	Description
X-Request-ID	UUID	Mandatory	Unique ID generated by the TPP to identify the request.
Consent-ID	String	Mandatory	Unique ID of the consent based on which the TPP has access to the accounts.
PSU-IP-Address	String	Mandatory	The IP address of the PSU. In the case of a call without PSU involvement, the TPP fills in 0.0.0.0.
PSU-Device-ID	String	Mandatory	The unique ID of the device used by the PSU. In the case of a call without PSU involvement, the TPP fills in no-psu-involved.
PSU-Device-Name	String	Mandatory	The name/model (generic) of the device from which the PSU connects. In the case of a call without PSU involvement, the TPP fills in no-psu-involved.
PSU-Geo-Location	String	Optional	The transmitted geographical location of the corresponding HTTP request between the PSU and TPP, if available.
Date	Datetime	Mandatory	The date and time when the request is made (RFC 7231).
Digest	String	Mandatory	It is included if and only if the "Signature" element is included in the request header.
Signature	String	Mandatory	Application-level signing of the request by the TPP.

Name	Type	Condition	Description
TPP-Signature-Certificate	String	Mandatory	The certificate used to sign the request, in base64 encoding. It must be included if a signature is present.

[TPP] Request Header Sample:

GET htttps://api.provider.com/v1/accounts/acc-123456789

X-Request-ID: 123e4567-e89b-12d3-a456-426614174000

Consent-ID: d6f9b8f4-4b10-4b9e-933b-ff9a24b5641f

PSU-IP-Address: 192.168.0.10

PSU-Device-ID: device-12345

PSU-Device-Name: ModelDevice X

Date: Wed, 11 Sep 2024 12:34:56 GMT

Digest: SHA-256=47DEQpj8HBSa+/TImW+5JCeuQeRkm5NMpJWZG3hSuFU=

Signature: keyId="SN= 4000000010FC01D520258AB15EAF, CA=CN=D-eSystemTrustIB, O=IP STISC 1003600096694, C=MD", algorithm="rsa-sha256",

headers="digest date x-request-id",

signature="Base64(RSA-SHA256(signing string))"

TPP-Signature-Certificate:
"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzKzT+I32ygAqDdZVfKYtDkWVZT7y
SP54ZXgH8dEUM6d9fKhs6DFiM9Do5slDDo7YwLjXU8Iq7C4eONHp+7u0z5LmvMyYnxgD0h1S7
F6T5gqaOQz3Qkm9bW2QY5M6Fh8/FivYpno3pzUNrzzTyAdIQ8MjbbJff7cDwDpwnFVgbQ6ZTx
Ym2CccovJQJuyfwO7ICtVjkkXq+FXWmZTfl2AfQwvMFuPRTlxjDLDBMOwDsYMBVBym8vSdzY
7AkDPylQtD/kTxMo+4toD+oLlo7mDs/qhvZXMnRPvE/JIE58xsiCBvUe36V1ht+WLidqk9iYxeAw
TbF7kZgxXjUGBYDrz/B4fqa2FqNzdsq2+LfsAk5cDBshXq1t/vmhty7TK09KPBrbDAjm9uDbf6zA0
ZSczX4rh7tBf3rc5BC+MBuLKgg1Pv9WgfWHi5BQ=="

[ASPSP] Response Header sample:

X-Request-ID: 123e4567-e89b-12d3-a456-426614174000

[ASPSP] Response Body Sample (Success - 200 OK):

{

"resourceId": "acc-123456789",

"iban": "MD21AAA000000022553456789",

"currency": "MDL",

```
 "product": "Cont Curent",
 "cashAccountType": "CACC",
"_links": {
        "balances": {
        "href": "/v1/accounts/ acc-123456789/balances"
        },
        "transactions": {
        "href": "/v1/accounts/ acc-123456789/transactions"
        }
    }
  }
```


[ASPSP] Response Body Sample withBalance (Success - 200 OK):


```
{
 "resourceId": "acc-123456789",
 "iban": "MD21AAA000000022553456789",
 "currency": "MDL",
 "product": "Cont MDL",
 "cashAccountType": "CACC",
    "balances": [
        {
          "balanceType": "interimAvailable",
           "balanceAmount": {
            "currency": "MDL",
            "amount": "99999.99"
          },
         "lastChangeDateTime": "2024-08-25T00:00:00Z"
       }
      ],
       "_links": {
          "transactions": {
```

```
            "href": "/v1/accounts/acc-123456789/transactions"
        }
    }
}
```

[ASPSP] Response Header Parameters:

Name	Type	Condition	Description
X-Request-ID	UUID	Mandatory	Unique ID generated by the TPP to identify the request.

[ASPSP] Parameters of Success Response:

Parameter	Type	Condition	Description
resourceId	String	Mandatory	The unique ID associated with each account.
iban	String	Mandatory	The account's IBAN.
currency	String	Mandatory	The account's currency (e.g., MDL, USD).
product	String	Mandatory	The type of product associated with the account (e.g., Current Account).
cashAccountType	String	Mandatory	The account type (e.g., CACC for current account, SVGS for savings account).
balances	Array	Mandatory (if withBalance)	The list of balances associated with the account.
balanceType	String	Mandatory (if withBalance)	The balance type (e.g., interimAvailable, expected).
balanceAmount	Object	Mandatory (if withBalance)	Details about the account balance.
balanceAmount.currency	String	Mandatory (if withBalance)	The account's currency (e.g., MDL, USD).
balanceAmount.amount	String	Mandatory (if withBalance)	Amount available in account.

Parameter	Type	Condition	Description
lastChangeDateTime	String	Mandatory (if withBalance)	Date and time of the last balance change.
_links	Object	Optional	Indicates the next steps that can be followed through the interface.

[ASPSP] **Response Sample (Error - 400 Bad Request):**


```
{
  "tppMessages": [
    {
      "category": "ERROR",
      "code": "CONSENT_INVALID",
      "text": "The consent provided is invalid or expired.",
      "path": "string"
    }
  ]
}
```


[ASPSP] **Parameters of Error Response:**

Parameter	Type	Condition	Description
tppMessages	Array	Mandatory	List of error messages generated by the server.
tppMessages.category	String	Mandatory	Message category (ERROR, WARNING).
tppMessages.code	String	Mandatory	Error code (e.g., FORMAT_ERROR).
tppMessages.text	String	Mandatory	Detailed description of the error (e.g., “Invalid IBAN format”).
tppMessages.path	String	Optional	Indicates the exact location of the error.

Method: Read Balance

Method type: GET /v1/accounts/{account-id}/balances

Description: This method allows the TPP to obtain the balances associated with a specific account, identified by the **accountId**. The returned data includes available

balances, booked balances, and other details related to the account's balance. This method is essential for providing Account Information Services (AIS).

[TPP] Path Parameters:

Name	Type	Condition	Description
account-id	String	Mandatory	The unique account ID assigned through the resourceId.

[TPP] Request Header Parameters:

Name	Type	Condition	Description
X-Request-ID	UUID	Mandatory	Unique ID generated by the TPP to identify the request.
Consent-ID	String	Mandatory	Unique ID of the consent based on which the TPP has access to the accounts.
PSU-IP-Address	String	Mandatory	The IP address of the PSU. In the case of a call without PSU involvement, the TPP fills in 0.0.0.0.
PSU-Device-ID	String	Optional	The unique ID of the device used by the PSU. In the case of a call without PSU involvement, the TPP fills in no-psu-involved.
PSU-Device-Name	String	Mandatory	The name/model (generic) of the device from which the PSU connects. In the case of a call without PSU involvement, the TPP fills in no-psu-involved.
PSU-Geo-Location	String	Optional	The transmitted geographical location of the corresponding HTTP request between the PSU and TPP, if available.
Date	Datetime	Mandatory	The date and time when the request is made (RFC 7231).
Digest	String	Mandatory	It is included if and only if the "Signature" element is included in the request header.
Signature	String	Mandatory	Application-level signing of the request by the TPP.
TPP-Signature-Certificate	String	Mandatory	The certificate used to sign the request, in base64 encoding. It must be included if a signature is present.

[TPP] Request Header Sample:

```
GET    https://api.provider.com/v1/accounts/acc-123456789/balances
```

X-Request-ID: 123e4567-e89b-12d3-a456-426614174000

Consent-ID: d6f9b8f4-4b10-4b9e-933b-ff9a24b5641f

PSU-IP-Address: 192.168.0.10

PSU-Device-ID: device-12345

PSU-Device-Name: ModelDevice X

Date: Wed, 11 Sep 2024 12:34:56 GMT

Digest: SHA-256=47DEQpj8HBSa+/TImW+5JCeuQeRkm5NMpJWZG3hSuFU=

Signature: keyId="SN= 4000000010FC01D520258AB15EAF, CA=CN=D-eSystemTrustIB, O=IP STISC 1003600096694, C=MD", algorithm="rsa-sha256",

headers="digest date x-request-id",

signature="Base64(RSA-SHA256(signing string))"

TPP-Signature-Certificate:
"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzKzT+I32ygAqDdZVfKYtDkWVZT7y
SP54ZXgH8dEUM6d9fKhs6DFiM9Do5slDDo7YwLjXU8Iq7C4eONHp+7u0z5LmvMyYnxgD0h1S7
F6T5gqaOQz3Qkm9bW2QY5M6Fh8/FivYpno3pzUNrzzTyAdIQ8MjbbJff7cDwDpwnFVgbQ6ZTx
Ym2CccovJQJuyfwO7ICtVjkkXq+FXWmZTfl2AfQwvMFuPRTlxjDLDBMOwDsYMBVBym8vSdzY
7AkDPylQtD/kTxMo+4toD+oLlo7mMtpTeDs/qhvZXMnRPvE/JIE58xsiCBvUe36V1ht+WLidqk9iY
xeAwTbF7kZgxXjUGBYDrz/B4fqa2FqNzdsq2+LfsAk5cDBshXq1t/vmhty7TK09KPBrbDAjm9uDb
f6zA0ZSczX4rh7tBf3rc5BC+Mv9WgfWHi5BQ=="

[ASPSP] Response Header sample:

X-Request-ID: 123e4567-e89b-12d3-a456-426614174000

[ASPSP] Response Body Sample (Success - 200 OK):


```
{
  "account": {
        "iban": "MD21AAA000000022553456789"
}
  "balances": [
   {
    "balanceType": "interimAvailable",
    "balanceAmount": {
     "currency": "MDL",
     "amount": "1000.00"
    },
```

```
    "lastChangeDateTime": "2024-09-11T12:34:56Z"
   },
   {
    "balanceType": "expected",
    "balanceAmount": {
     "currency": "MDL",
     "amount": "200.00"
    },
    "lastChangeDateTime": "2024-09-10T15:00:00Z"
   }
 ]
}
```


[ASPSP] Response Header Parameters:

Name	Type	Condition	Description
X-Request-ID	UUID	Mandatory	Unique ID generated by TPP to identify the request.

[ASPSP] Parameters of Success Response:

Parameter	Type	Condition	Description
balances	Array	Mandatory	The list of balances associated with the account.
balanceType	String	Mandatory	The balance type (e.g., interimAvailable, expected).
balanceAmount	Object	Mandatory	Details about the account balance.
balanceAmount.currency	String	Mandatory	The account's currency (e.g., MDL).
balanceAmount.amount	String	Mandatory	Amount available in account.
lastChangeDateTime	String	Mandatory	Date and time of the last balance change.

[ASPSP] Response Sample (Error - 400 Bad Request):


```
{
```

```
 "tppMessages": [
  {
   "category": "ERROR",
   "code": "CONSENT_INVALID",
   "text": "string",
   "path": "string"
  }
 ]
}
```


[ASPSP] Parameters of Error Response:

Parameter	Type	Condition	Description
tppMessages	Array	Mandatory	List of error messages generated by the server.
tppMessages.category	String	Mandatory	Message category (ERROR, WARNING).
tppMessages.code	String	Mandatory	Error code (e.g., FORMAT_ERROR).
tppMessages.text	String	Mandatory	Detailed description of the error (e.g., "Invalid IBAN format").
tppMessages.path	String	Optional	Indicates the exact location of the error.

Method Read transaction list of an account

Method type: GET /v1/accounts/ {account-id}/transactions

Description: This method allows the TPP to obtain a list of transactions associated with a specific account, identified by **accountId**. The returned transactions include both booked transactions and pending transactions. This method is essential for providing Account Information Services (AIS).

To determine the type of payment, whether it is an incoming payment (credit) or an outgoing payment (payment made), it is based on the information below:

- The presence of the creditorAccount field → outgoing payment (money goes to the creditor) → Debit payment
- The presence of the debtorAccount field → incoming payment (money comes from the debtor) → Credit payment

[TPP] Path Parameters:

Name	Type	Condition	Description
account-id	String	Mandatory	The unique account ID assigned through the resourceId.

[TPP] Query Parameters:

Name	Type	Condition	Description
dateFrom	ISODate	Optional	Start date for filtering transactions (format: YYYY-MM-DD).
dateTo	ISODate	Optional	End date for filtering transactions (format: YYYY-MM-DD).
bookingStatus	String	Mandatory	The status of the transactions to be returned (booked, pending, or both).
withBalance	Boolean	Optional	If set to true, the response will also include the account balance.

[TPP] Request Header Parameters:

Name	Type	Condition	Description
X-Request-ID	UUID	Mandatory	Unique ID generated by the TPP to identify the request.
Consent-ID	String	Mandatory	Unique ID of the consent based on which the TPP has access to the accounts.
PSU-IP-Address	String	Mandatory	The IP address of the PSU. In the case of a call without PSU involvement, the TPP fills in 0.0.0.0.
PSU-Device-ID	String	Mandatory	The unique ID of the device used by the PSU. In the case of a call without PSU involvement, the TPP fills in no-psu-involved.
PSU-Device-Name	String	Mandatory	The name/model (generic) of the device from which the PSU connects. In the case of a call without PSU involvement, the TPP fills in no-psu-involved.
PSU-Geo-Location	String	Optional	The transmitted geographical location of the corresponding HTTP request between the PSU and TPP, if available.
Date	String	Mandatory	The date and time when the request is made (RFC 7231).

Name	Type	Condition	Description
Digest	String	Mandatory	It is included if and only if the "Signature" element is included in the request header.
Signature	String	Mandatory	Application-level signing of the request by the TPP.
TPP-Signature-Certificate	String	Mandatory	The certificate used to sign the request, in base64 encoding. It must be included if a signature is present.

[TPP] Request Header Sample:

GET / https://api.provider.com/v1/accounts/acc-123456789/transactions?dateFrom=2024-01-01&dateTo=2024-09-01&bookingStatus=booked

X-Request-ID: 123e4567-e89b-12d3-a456-426614174000

Consent-ID: d6f9b8f4-4b10-4b9e-933b-ff9a24b5641f

PSU-IP-Address: 192.168.0.10

PSU-Device-ID: device-12345

PSU-Device-Name: ModelDevice X

Date: Wed, 11 Sep 2024 12:34:56 GMT

Digest: SHA-256=47DEQpj8HBSa+/TImW+5JCeuQeRkm5NMpJWZG3hSuFU=

Signature: keyId="SN= 4000000010FC01D520258AB15EAF, CA=CN=D-eSystemTrustIB, O=IP STISC 1003600096694, C=MD", algorithm="rsa-sha256",

headers="digest date x-request-id",

signature="Base64(RSA-SHA256(signing string))"

TPP-Signature-Certificate:
"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzKzT+I32ygAqDdZVfKYtDkWVZT7y
SP54ZXgH8dEUM6d9fKhs6DFiM9Do5slDDo7YwLjXU8Iq7C4eONHp+7u0z5LmvMyYnxgD0h1S7
F6T5gqaOQz3Qkm9bW2QY5M6Fh8/FivYpno3pzUNrzzTyAdIQ8MjbbJff7cDwDpwnFVgbQ6ZTx
Ym2CccovJQJuyfwO7ICtVjkkWmZTfl2AfQwvMFuPRTlxjDLDBMOwDsYMBVBym8vSdzY7AkDP
ylQtD/kTxMo+4toD+oLlo7mMtpTeDs/qhvZXMnRPvE/JIE58xsiCBvUe36V1ht+WLidqk9iYxeAw
TbF7kZgxXjUGBYDrz/B4fqa2FqNzdsq2+LfsAk5cDBshXq1t/vmhty7TK09KPBrbDAjm9uDbf6zA0
ZSczX4rh7tBf3rc5BC+Mv9WgfWHi5BQ=="

[ASPSP] Response Header sample:

X-Request-ID: 123e4567-e89b-12d3-a456-426614174000

Content-Type: application/json

[ASPSP] Response Body Sample (Success - 200 OK):

{

```
"account": {
  "iban": "MD21AAA000000022553456789",
  "currency": "MDL"
},
"transactions": {
  "booked": [
    {
      "transactionId": "tx-123",
      "creditorName": " Ion Popescu ",
      "creditorAccount": {
        "iban": "MD21AAA000000022553456789"
      },
      "transactionAmount": {
        "currency": "MDL",
        "amount": "500.00"
      },
      "bookingDate": "2024-09-01",
      "valueDate": "2024-09-01",
      "remittanceInformationUnstructured": "Plată Factura 123"
    },
    {
      "transactionId": "tx-1234",
      "debtorName": "Petru Popescu",
      "debtorAccount": {
        "iban": "MD32AAA000000022663456789"
      },
      "transactionAmount": {
        "currency": "MDL",
        "amount": "2500.00"
      },
```

```
      "bookingDate": "2024-09-15",
      "valueDate": "2024-09-18",
      "remittanceInformationUnstructured": "P2P"
    }
  ],
  "pending": [
    {
      "transactionId": "tx-456",
      "creditorName": " Ion Popescu ",
      "creditorAccount": {
        "iban": "MD21AAA000000022553456789"
      },
      "transactionAmount": {
        "currency": "MDL",
        "amount": "200.00"
      },
      "valueDate": "2024-09-01",
      "remittanceInformationUnstructured": "Plată Factura 123"
    }
  ]
 }
}
```


[ASPSP] Response Body Sample withBalance (Success - 200 OK):


```
{
 "account": {
  "iban": "MD21AAA000000022553456789",
  "currency": "MDL"
 },
 "transactions": {
  "booked": [
```

```
    {
      "transactionId": "tx-123",
      "creditorName": "Ion Popescu",
      "creditorAccount": {
        "iban": "MD21AAA000000022553456789"
      },
      "transactionAmount": {
        "currency": "MDL",
        "amount": "500.00"
      },
      "bookingDate": "2024-09-01",
      "valueDate": "2024-09-01",
      "remittanceInformationUnstructured": "Plată Factura 123"
    }
  ]
 },
 "balances": [
  {
   "balanceType": "interimAvailable",
   "balanceAmount": {
     "currency": "MDL",
     "amount": "99499.99"
   },
   "lastChangeDateTime": "2024-09-01T12:00:00Z"
  }
 ]
}
```


[ASPSP] Response Header Parameters:

Name	Type	Condition	Description
X-Request-ID	UUID	Mandatory	Unique ID generated by the TPP to identify the request.
Content-Type	String	Mandatory	Specifies that the response body format is application/json.

[ASPSP] **Parameters of Success Response:**

Parameter	Type	Condition	Description
account	Object	Mandatory	Account information associated with the transactions.
account.iban	String	Mandatory	The account's IBAN.
account.currency	String	Mandatory	The account's currency.
transactions	Object	Mandatory	Transaction information from the query parameters.
transactions.booked	Array	Mandatory	Booked transactions.
transactions.pending	Array	Mandatory	Pending transactions.
transactionId	String	Mandatory	The unique transaction ID.
bookingDate	String	Mandatory	The transaction booking date.
valueDate	String	Mandatory	The transaction execution date.
transactionAmount	Object	Mandatory	Contains the amount and currency of the transaction.
transactionAmount.currency	String	Mandatory	Currency of the transaction.
transactionAmount.amount	String	Mandatory	Transaction amount.
creditorName	String Max70Text	Mandatory (if Debit)	Name of the creditor.
creditorAccount	Object	Mandatory (if Debit and held by ASPSP)	The creditor's account.

Parameter	Type	Condition	Description
creditorAccount.iban	String	Mandatory (if Debit and held by ASPSP)	Creditor's IBAN.
debtorName	String	Mandatory (if Credit)	Name of the debtor.
debtorAccount	Object	Mandatory (if Credit and held by ASPSP)	The debtor's account.
debtorAccount.iban	String	Mandatory (if Credit and held by ASPSP)	The debtor's IBAN.
remittanceInformationUnstructured	String Max420Text [0-9a-zA-Z/\-\?:\(\)\.,'\+]{1,35}	Mandatory	Unstructured payment information (e.g., invoice details). If the ASPSP provides such information in its application, it is required to provide it to the TPP as well.
balances	Array	Mandatory (if contains withBalance)	The list of balances associated with the account.
balanceType	String	Mandatory (if contains withBalance)	The balance type (e.g., interimAvailable, expected).
balanceAmount	Object	Mandatory (if contains withBalance)	Details about the account balance.
balanceAmount.currency	String	Mandatory (if contains withBalance)	The account's currency (e.g., MDL, USD).
balanceAmount.amount	String	Mandatory (if contains withBalance)	Amount available in account.

Parameter	Type	Condition	Description
lastChangeDateTime	String	Mandatory (if contains withBalance)	Date and time of the last balance change.

[ASPSP] **Response Sample (Error - 400 Bad Request):**


```
{
  "tppMessages": [
    {
      "category": "ERROR",
      "code": "FORMAT_ERROR",
      "text": "string",
      "path": "string"
    }
  ]
}
```


[ASPSP] **Parameters of Error Response:**

Parameter	Type	Condition	Description
tppMessages	Array	Mandatory	List of error messages generated by the server.
tppMessages.category	String	Mandatory	Message category (ERROR, WARNING).
tppMessages.code	String	Mandatory	Error code (e.g., FORMAT_ERROR).
tppMessages.text	String	Mandatory	Detailed description of the error (e.g., "Invalid IBAN format").
tppMessages.path	String	Optional	Indicates the exact location of the error.

3. Payments Endpoint

Method: Payment initiation

Method type: POST /v1/{payment-service}/{payment-product}

Description: This method initiates a payment based on the chosen product type, which currently includes domestic and instant payments, as defined by the national scheme in the Republic of Moldova. Each type of payment is specified by {payment-product}.

[TPP] Path Parameters:

Name	Type	Condition	Description
payment-service	String	Mandatory	The type of payment service (can be: payments, bulk-payments, periodic-payments).
payment-product	String	Mandatory	The type of product used (e.g., domestic-credit-transfers-md or instant-credit-transfers-md).

[TPP] Request Header Parameters:

Name	Type	Condition	Description
X-Request-ID	UUID	Mandatory	Unique ID generated by TPP to identify the request. It is mandatory for uniqueness verification by the ASPSP.
Content-type	String	Mandatory	Specifies the request body format application/json.
PSU-IP-Address	String	Mandatory	The IP address of the PSU.
PSU-Device-ID	String	Mandatory	The device ID used by the PSU.
PSU-Device-Name	String	Mandatory	The name/model (generic) of the device from which the PSU connects.
PSU-Geo-Location	Geo Location	Mandatory	The geographical location of the PSU.
TPP-Redirect-URI	String	Mandatory	The redirect URI to the TPP after the payment is completed.
TPP-Nok-Redirect-URI	String	Optional	The redirect URI in case of an error.
Date	Datetime	Mandatory	The date and time when the request is made (RFC 7231).
Digest	String	Mandatory	It is included only if and only if the “Signature” element is included in the request header.
Signature	String	Mandatory	Application-level signing of the request by the TPP.
TPP-Signature-Certificate	String	Mandatory	The certificate used to sign the request, in base64 encoding. It must be included if a signature is present.

Example of initiating a payment through the product: domestic-credit-transfers-md

[TPP] Request Header Sample:

GET https://api.provider.com/v1/payments/ domestic-credit-transfers-md

X-Request-ID: 123e4567-e89b-12d3-a456-426614174000

Content-Type: application/json

PSU-IP-Address: 192.168.0.10

PSU-Device-ID: device-12345

PSU-Geo-Location: GEO:47.046399;28.762064

TPP-Redirect-URI: https://tpp-example.md/redirect

TPP-Nok-Redirect-URI: https://tpp-example.md/redirect-failure

Date: Wed, 11 Sep 2024 12:34:56 GMT

Digest: SHA-256=i/AwfzfBZztOinTJq+ANgtvyxF4ukmQjGM+Ae+5Twhs=

Signature: keyId="SN= 4000000010FC01D520258AB15EAF, CA=CN=D-eSystemTrustIB, O=IP STISC 1003600096694, C=MD", algorithm="rsa-sha256",

headers="digest date x-request-id",

signature="Base64(RSA-SHA256(signing string))"

TPP-Signature-Certificate:
"MIIBIjANBgkqhkiQEFAAOCAQ8AMIIBCgKCAQEAzKzT+I32ygAqDdZVfKYtDkWVZT7ySP54ZXg
H8dEUM6d9fKhs6DFiM9Do5slDDo7YwLjXU8Iq7C4eONHp+7u0z5LmvMyYnxgD0h1S7F6T5gqa
OQz3Qkm9bW2QY5M6Fh8/FivYpno3pzUNrzzTyAdIQ8MjbbJff7cDwDpwnFVgbQ6ZTxYm2Ccco
vJQJuyfwO7ICtVjkkWmZTfl2AfQwvMFuPRTlxjDLDBMOwDsYMBVBym8vSdzY7AkDPylQtD/k
TxMo+4toD+oLlo7mMtpTeDs/qhvZXMnRPvE/JIE58xsiCBvUe36V1ht+WLidqk9iYxeAwTbF7kZg
xXjUGBYDrz/B4fqa2FqNzdsq2+LfsAk5cDBshXq1t/vmhty7TK09KPBrbDAjm9uDbf6zA0ZSczX4r
h7tBf3rc5BC+Mv9WgfWHi5BQ=="

[TPP] Request body Sample:

{

"endToEndIdentification": "cc5a8022-5e71-460e-82fa-ab0be1997a5",

"instructedAmount": {

"currency": "MDL",

"amount": "1000.00"

},

"debtorAccount": {

"iban": "MD12AA000001100032130935"

```
 },
 "creditorName": "Comerciant X",
 "creditorId": "2002002002002",
 "creditorOrgId": "ABCDEFGHI1ABCDFD1212",
 "creditorCtryOfRes": "MD",
 "creditorAccount": {
  "iban": "MD24AA000001100032130935"
 },
 "instructionPriority": "NORM",
 "remittanceInformationUnstructured": "Plata facturii #123"
}
```


[TPP] Parameters of body request:

Parameter	Type	Condition	Description
endToEndIdentification	String Max35Text	Mandatory	The unique payment ID from the TPP for tracking the payment.
debtorAccount	Object	Optional	Information about the debtor's account.
debtorAccount.iban	String	Optional	The IBAN of the debtor's account.
instructedAmount	Object	Mandatory	The amount instructed for transfer.
instructedAmount.currency	String	Mandatory	The account's currency (e.g., MDL).
instructedAmount.amount	String	Mandatory	The payment amount.
creditorName	String Max70Text	Mandatory	The beneficiary/creditor's name.
creditorId	String Max35Text	Mandatory	The beneficiary's ID (IDNP/IDNO or non-resident ID).

Parameter	Type	Condition	Description
endToEndIdentification	String Max35Text	Mandatory	The unique payment ID from the TPP for tracking the payment.
creditorOrgId	String Max20Text [A-Z0-9]	Optional	The legal entity identifier (LEI) – a unique 20-character alphanumeric reference code, based on the ISO 17442 standard, assigned to a legal entity.
creditorCtryOfRes	String Text [A-Z]{2,2}	Mandatory	The beneficiary's country code (ISO 3166-1).
creditorAccount	Oject	Mandatory	Information about the creditor's account.
creditorAccount.iban	String	Mandatory	The IBAN of the creditor's account.
instructionPriority	String	Mandatory	The type of transfer, indicating the transfer regime. NORM – for regular payments URGT – for urgent payments.
remittanceInformationUnstructured	String Max420Text [0-9a-zA-Z/\-\?:\(\)\.,'\+]{1,35}	Optional (in the case of A2A mandatory for the ASPSP)	Additional payment information (free text).

Example of initiating a payment through the product: instant-credit-transfers-md

[TPP] Request Header Sample:

```
GET https://api.provider.com/v1/payments/ instant- credit-transfers-md

X-Request-ID: 123e4567-e89b-12d3-a456-426614174000

Content-Type: application/json

PSU-IP-Address: 192.168.0.10

PSU-Device-ID: device-12345
```

PSU-Device-Name: ModelDevice X

PSU-Geo-Location: GEO:47.046399;28.762064

TPP-Redirect-URI: https://tpp-example.md/redirect

TPP-Nok-Redirect-URI: https://tpp-example.md/redirect-failure

Date: Wed, 11 Sep 2024 12:34:56 GMT

Digest: SHA-256=+8pwEuPygqm1u33m0Y0LCSlsU8UhTgYDhMZhoi871rU=

Signature: keyId="SN= 4000000010FC01D520258AB15EAF, CA=CN=D-eSystemTrustIB, O=IP STISC 1003600096694, C=MD", algorithm="rsa-sha256",

headers="digest date x-request-id",

signature="Base64(RSA-SHA256(signing string))"

TPP-Signature-Certificate:
"MIIBIjANBgkqhkiQEFAAOCAQ8AMIIBCgKCAQEAzKzT+I32ygAqDdZVfKYtDkWVZT7ySP54ZXg
H8dEUM6d9fKhs6DFiM9Do5slDDo7YwLjXU8Iq7C4eONHp+7u0z5LmvMyYnxgD0h1S7F6T5gqa
OQz3Qkm9bW2QY5M6Fh8/FivYpno3pzUNrzzTyAdIQ8MjbbJff7cDwDpwnFVgbQ6ZTxYm2Ccco
vJQJuyfwO7ICtVjkkWmZTfl2AfQwvMFuPRTlxjDLDBMOwDsYMBVBym8vSdzY7AkDPylQtD/k
TxMo+4toD+oLlo7mMtpTeDs/qhvZXMnRPvE/JIE58xsiCBvUe36V1ht+WLidqk9iYxeAwTbF7kZg
xXjUGBYDrz/B4fqa2FqNzdsq2+LfsAk5cDBshXq1t/vmhty7TK09KPBrbDAjm9uDbf6zA0ZSczX4r
h7tBf3rc5BC+Mv9WgfWHi5BQ==",

[TPP] Request body Sample:

```
{
 "endToEndIdentification": "d14c3e75-8a2f-4e93-b3ca-ec4fd7128b9e",
 "instructedAmount": {
  "currency": "MDL",
  "amount": "1000.00"
 },
 "debtorAccount": {
  "iban": "MD12AA000001100032130935"
 },
 "creditorAccount": {
  "msisdn ": "37399000000"
 },
 "purposeCode": "201",
 "remittanceInformationUnstructured": "Transfer P2P"
```

}

[TPP] Parameters of body request:

Parameter	Type	Condition	Description
endToEndIdentification	String Max35Text	Mandatory	The unique payment ID from the TPP for tracking the payment.
debtorAccount	Object	Optional	Information about the debtor's account.
debtorAccount.iban	String	Mandatory	The IBAN of the debtor's account.
instructedAmount	Object	Mandatory	The amount instructed for transfer.
instructedAmount.currency	String	Mandatory	The account's currency (e.g., MDL).
instructedAmount.amount	String	Mandatory	The payment amount.
creditorAccount	Object	Mandatory	Information about the creditor's account.
creditorAccount.msisdn	String	Mandatory	Alias of the creditor's account.
purposeCode	String	Mandatory	The TCC code, according to the NBM documentation. Currently, only TCC 201 will be used for P2P payments.
remittanceInformationUnstructured	String Max420Text [0-9a-zA-Z/\-\?:\(\)\.,'\+]{1,35}	Optional	Additional payment information (free text).

[ASPSP] Response Header sample:

```
Content-Type: application/json

X-Request-ID: 123e4567-e89b-12d3-a456-426614174000

ASPSP-SCA-Approach: REDIRECT
```

```
Date: Wed, 11 Sep 2024 12:34:58 GMT

Location: “/v1/payments/ domestic-credit-transfers-md/MD123456789”
```

[ASPSP] Response Sample (201 Created):

```
{
 "transactionStatus": "RCVD",
 "paymentId": "MD123456789",
 "_links": {
  "scaRedirect": {
   "href": "www.mybankapp.md//authentication/ a7c3e9f8-1a44-4cd3-83ab-4f29d1f9e8c7 "
  },
  "self": {
   "href": "/v1/payments/MD123456789"
  },
  "status": {
   "href": "/v1/payments/ MD123456789/status"
  }
 }
}
```

[ASPSP] Parameters of Success Response:

Parameter	Type	Condition	Description
transactionStatus	String	Mandatory	The status of the transaction.
paymentId	String	Mandatory	Unique payment identifier.
_links	String	Mandatory	Link for the details of the initiated transaction.

[ASPSP] Response Sample (Error or Warning):

```
{
 "tppMessages": [
  {
   "category": "ERROR",
   "code": "FORMAT_ERROR",
```

```
  "path": "string",
  "text": "string"
}
```

[ASPSP] Parameters of Error Response:

Parameter	Type	Condition	Description
tppMessages	Array	Mandatory	List of error messages generated by the server.
tppMessages.category	String	Mandatory	Message category (ERROR, WARNING).
tppMessages.code	String	Mandatory	Error code (e.g., FORMAT_ERROR).
tppMessages.text	String	Mandatory	Detailed description of the error (e.g., “Invalid IBAN format”).
tppMessages.path	String	Optional	Indicates the exact location of the error.

Method: Get payment information

Method type: GET /v1/{payment-service}/{payment-product}/{paymentId}

Description: This method allows viewing the details of an initiated payment using the unique payment identifier ({paymentId}) for the specified payment type ({payment-product}).

[TPP] Path Parameters:

Name	Type	Condition	Description
payment-service	String	Mandatory	The type of payment service (can be: payments, bulk-payments, periodic-payments).
payment-product	String	Mandatory	The type of product used (e.g., domestic-credit-transfers-md or instant-credit-transfers-md).
paymentId	String	Mandatory	The identifier of the initiated payment.

[TPP] Request Header Parameters:

Name	Type	Condition	Description
X-Request-ID	UUID	Mandatory	Unique ID generated by the TPP to identify the request.

Name	Type	Condition	Description
PSU-IP-Address	String	Mandatory	The IP address of the PSU. In the case of a call without PSU involvement, the TPP fills in 0.0.0.0.
PSU-Device-ID	String	Mandatory	The unique ID of the device used by the PSU. In the case of a call without PSU involvement, the TPP fills in no-psu-involved.
PSU-Device-Name	String	Mandatory	The name/model (generic) of the device from which the PSU connects. In the case of a call without PSU involvement, the TPP fills in no-psu-involved.
PSU-Geo-Location	Geo Location	Optional	The transmitted geographical location of the corresponding HTTP request between the PSU and TPP, if available.
Date	Datetime	Mandatory	The date and time when the request is made (RFC 7231).
Digest	String	Mandatory	It is included if and only if the "Signature" element is included in the request header.
Signature	String	Mandatory	Application-level signing of the request by the TPP.
TPP-Signature-Certificate	String	Mandatory	The certificate used to sign the request, in base64 encoding. It must be included if a signature is present.

[TPP] Request Sample:

GET https://api.provider.com/v1/payments/ domestic-credit-transfers-md/MD123456789

X-Request-ID: 123e4567-e89b-12d3-a456-426614174000

PSU-IP-Address: 192.168.0.10

PSU-Device-ID: device-12345

PSU-Device-Name: ModelDevice X

Date: Wed, 11 Sep 2024 12:34:56 GMT

Digest: SHA-256=47DEQpj8HBSa+/TImW+5JCeuQeRkm5NMpJWZG3hSuFU=

Signature: keyId="SN= 4000000010FC01D520258AB15EAF, CA=CN=D-eSystemTrustIB, O=IP STISC 1003600096694, C=MD", algorithm="rsa-sha256",

headers="digest date x-request-id",

signature="Base64(RSA-SHA256(signing string))"

TPP-Signature-Certificate:
"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzKzT+I32ygAqDdZVfKYtDkWVZT7y
SP54ZXgH8dEUM6d9fKhs6DFiM9Do5slDDo7YwLjXU8Iq7C4eONHp+7u0z5LmvMyYnxgD0h1S7
F6T5gqaOQz3Qkm9bW2QY5M6Fh8/FivYpno3pzUNrzzTyAdIQ8MjbbJff7cDwDpwnFVgbQ6ZTx
Ym2CccovJQJuyfwO7ICtVjkkWmZTfl2AfQwvMFuPRTlxjDLDBMOwDsYMBVBym8vSdzY7AkDP
ylQtD/kTxMo+4toD+oLlo7mMtpTeDs/qhvZXMnRPvE/JIE58xsiCBvUe36V1ht+WLidqk9iYxeAw
TbF7kZgxXjUGBYDrz/B4fqa2FqNzdsq2+LfsAk5cDBshXq1t/vmhty7TK09KPBrbDAjm9uDbf6zA0
ZSczX4rh7tBf3rc5BC+Mv9WgfWHi5BQ==",

[ASPSP] Response Sample (200 OK) (domestic-credit-transfers-md):


```
{
 "paymentId": "MD123456789",
 "debtorAccount": {
  "iban": "MD12AA000001100032130935"
 },
 "instructedAmount": {
  "currency": "MDL",
  "amount": "1000.00"
 },
 "creditorAccount": {
  "iban": "MD24AA000001100032130935"
 },
 "creditorName": "Nume Beneficiar",
 "remittanceInformationUnstructured": "Plata facturii #123",
 "transactionStatus": "ACCP",
 "transactionFees": {
  "currency": "MDL",
  "amount": "5.00"
 }
}
```


[ASPSP] Parameters of Success Response (domestic-credit-transfers-md):

Parameter	Type	Condition	Description
paymentId	String	Mandatory	The unique payment identifier.
transactionStatus	String	Mandatory	The current status of the payment.
debtorAccount	Object	Mandatory	The debtor's account.
debtorAccount.iban	String	Mandatory	The IBAN of the debtor's account.
instructedAmount	Object	Mandatory	The instructed payment amount (currency and value).
instructedAmount.currency	String	Mandatory	The account's currency (e.g., MDL).
instructedAmount.amount	String	Mandatory	The payment amount.
creditorAccount	Object	Mandatory	The creditor's account.
creditorAccount.iban	String	Mandatory	The IBAN of the creditor's account.
creditorName	String Max70Text	Mandatory	The name of the payment beneficiary.
remittanceInformationUnstructured	String Max420Text [0-9a-zA-Z/\-\?:\(\)\.,'\+]{1,35}	Optional	Additional payment information.
transactionStatus	String	Mandatory	The payment status.
transactionFees	Object	Optional	The transaction fees.
transactionFees.currency	String	Optional	The currency of the fee (e.g., MDL).
transactionFees.amount	String	Optional	The fee amount.

[ASPSP] Response Sample for instant-credit-transfers-md (200 OK):


```
{
 "paymentId": " MD123456789",
 "debtorAccount": {
  "iban": "MD12AA000001100032130935"
 },
 "instructedAmount": {
  "currency": "MDL",
  "amount": "1000.00"
 },
 "creditorAccount": {
  "msisdn": "37399000000"
 },
 "remittanceInformationUnstructured": "Plata P2P",
 "transactionStatus": "ACTC",
 "transactionFees": {
  "currency": "MDL",
  "amount": "5.00"
 }
```


[ASPSP] Parameters of Success Response (domestic-credit-transfers-md):

Parameter	Type	Condition	Description
paymentId	String	Mandatory	The unique payment identifier.
transactionStatus	String	Mandatory	The current status of the payment.
debtorAccount	Object	Mandatory	The debtor's account.
debtorAccount.iban	String	Mandatory	The IBAN of the debtor's account.

Parameter	Type	Condition	Description
instructedAmount	Object	Mandatory	The instructed payment amount (currency and value).
instructedAmount.currency	String	Mandatory	The account's currency (e.g., MDL).
instructedAmount.amount	String	Mandatory	The payment amount.
creditorAccount	Object	Mandatory	The creditor's account.
creditorAccount.iban	String	Mandatory	The IBAN of the creditor's account.
creditorName	String Max70Text	Mandatory	The name of the payment beneficiary.
remittanceInformationUnstructured	String Max420Text [0-9a-zA-Z/\-\?:\(\)\.,'\+]{1,35}	Optional	Additional payment information.
transactionStatus	String	Mandatory	The payment status.
transactionFees	Object	Optional	The transaction fees.
transactionFees.currency	String	Optional	The currency of the fee (e.g., MDL).
transactionFees.amount	String	Optional	The fee amount.

[ASPSP] Response Sample (Error or Warning):


```
{
  "tppMessages": [
    {
      "category": "ERROR",
      "code": "FORMAT_ERROR",
      "text": "string",
```

```
  "path": "string",
}
```

[ASPSP] Parameters of Error Response:

Parameter	Type	Condition	Description
tppMessages	Array	Mandatory	List of error messages generated by the server.
tppMessages.category	String	Mandatory	Message category (ERROR, WARNING).
tppMessages.code	String	Mandatory	Error code (e.g., FORMAT_ERROR).
tppMessages.text	String	Mandatory	Detailed description of the error (e.g., “Invalid IBAN format”).
tppMessages.path	String	Optional	Indicates the exact location of the error.

Method: Get payment status

Method type: GET /v1/{payment-service}/{payment-prodyuct}/{paymentId}/status

Description: This method returns the current status of an initiated payment, identified by {paymentId}, for the specified payment product type ({payment-product}).

[TPP] Path Parameters:

Name	Type	Condition	Description
payment-service	String	Mandatory	The type of payment service (can be: payments, bulk-payments, periodic-payments).
payment-product	String	Mandatory	The type of product used (e.g., domestic-credit-transfers-md or instant-credit-transfers-md).
paymentId	String	Mandatory	The identifier of the initiated payment.

[TPP] Request Header Parameters:

Name	Type	Condition	Description
X-Request-ID	UUID	Mandatory	Unique ID generated by the TPP to identify the request.
PSU-IP-Address	String	Mandatory	The IP address of the PSU. In the case of a call without PSU involvement, the TPP fills in 0.0.0.0.

Name	Type	Condition	Description
PSU-Device-ID	String	Mandatory	The ID of the device used by the PSU. In the case of a call without PSU involvement, the TPP fills in no-psu-involved.
PSU-Device-Name	String	Mandatory	The name/model (generic) of the device from which the PSU connects. In the case of a call without PSU involvement, the TPP fills in no-psu-involved.
PSU-Geo-Location	Geo Location	Optional	The geographical location of the PSU.
Date	Datetime	Mandatory	The date and time when the request is made (RFC 7231).
Digest	String	Mandatory	It is included only if and only if the "Signature" element is included in the request header.
Signature	String	Mandatory	Application-level signing of the request by the TPP.
TPP-Signature-Certificate	String	Mandatory	The certificate used to sign the request, in base64 encoding. It must be included if a signature is present.

[TPP] Request Header Sample:

GET https://api.provider.com/v1/payments/domestic-credit-transfers-md/MD123456789/status

X-Request-ID: 123e4567-e89b-12d3-a456-426614174000

PSU-IP-Address: 192.168.0.10

PSU-Device-ID: device-12345

PSU-Device-Name: ModelDevice X

Date: Wed, 11 Sep 2024 12:34:56 GMT

Digest: SHA-256=47DEQpj8HBSa+/TImW+5JCeuQeRkm5NMpJWZG3hSuFU=

Signature: keyId="SN= 4000000010FC01D520258AB15EAF, CA=CN=D-eSystemTrustIB, O=IP STISC 1003600096694, C=MD", algorithm="rsa-sha256",

headers="digest date x-request-id",

signature="Base64(RSA-SHA256(signing string))"

TPP-Signature-Certificate: "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzKzT+I32ygAqDdZVfKYtDkWVZT7y

SP54ZXgH8dEUM6d9fKhs6DFiM9Do5slDDo7YwLjXU8Iq7C4eONHp+7u0z5LmvMyYnxgD0h1S7
F6T5gqaOQz3Qkm9bW2QY5M6Fh8/FivYpno3pzUNrzzTyAdIQ8MjbbJff7cDwDpwnFVgbQ6ZTx
Ym2CccovJQJuyfwO7ICtVjkkWmZTfl2AfQwvMFuPRTlxjDLDBMOwDsYMBVBym8vSdzY7AkDP
ylQtD/kTxMo+4toD+oLlo7mMtpTeDs/qhvZXMnRPvE/JIE58xsiCBvUe36V1ht+WLidqk9iYxeAw
TbF7kZgxXjUGBYDrz/B4fqa2FqNzdsq2+LfsAk5cDBshXq1t/vmhty7TK09KPBrbDAjm9uDbf6zA0
ZSczX4rh7tBf3rc5BC+Mv9WgfWHi5BQ=="

[ASPSP] Response Header sample:

```
X-Request-ID: 123e4567-e89b-12d3-a456-426614174000
```

[ASPSP] Response Body Sample (200 OK):

```
{
  "transactionStatus": "ACCP"
}
```

[ASPSP] Response Header Parameters:

Name	Type	Condition	Description
X-Request-ID	UUID	Mandatory	Unique ID generated by the TPP to identify the request.

[ASPSP] Parameters of Success Response:

Parameter	Type	Condition	Description
transactionStatus	String	Mandatory	The current status of the payment.

[ASPSP] Response Sample (Error or Warning):

```
{
  "tppMessages": [
    {
      "category": "ERROR",
      "code": "FORMAT_ERROR",
      "path": "string",
      "text": "string"
    }
```

[ASPSP] Parameters of Error Response:

Parameter	Type	Condition	Description
tppMessages	Array	Mandatory	List of error messages generated by the server.
tppMessages.category	String	Mandatory	Message category (ERROR, WARNING).
tppMessages.code	String	Mandatory	Error code (e.g., FORMAT_ERROR).
tppMessages.text	String	Mandatory	Detailed description of the error (e.g., “Invalid IBAN format”).
tppMessages.path	String	Optional	Indicates the exact location of the error.

Annex No 2

The types of responses and standardized error codes

1. Consent Status

Code	Description
received	The consent data has been received and is technically correct. The data is not yet authorized.
rejected	The consent data has been rejected, for example, due to the absence of a successful authorization.
partiallyAuthorised	The consent requires multi-level authorization, but not all mandatory authorizations have been completed yet.
valid	The consent is accepted and valid for GET calls for account data and other operations specified in the consent object.
revokedByPsu	The consent has been revoked by the PSU to the ASPSP.
expired	The consent has expired.
terminatedByTpp	The corresponding TPP has revoked the consent by applying the DELETE method to the consent resource.

2. Account Access

Attribute	Type	Description
accounts	Array	Request for account details.
balances	Array	Request for account balances.
transactions	Array	Request for account transactions.
additionalInformation	Structurat	Request for additional structured information.
availableAccounts	String	Allowed values: "allAccounts" and "allAccountsWithOwnerName".
availableAccountsWithBalance	String	Allowed values: "allAccounts" and "allAccountsWithOwnerName".

3. Account Reference

Attribute	Type	Description
iban	IBAN	IBAN identifier of the account.
bban	BBAN	BBAN code, used for accounts that do not have an IBAN.
pan	PAN	Account number for cards, used instead of IBAN or BBAN where applicable.
maskedPan	Masked PAN	Masked PAN, used to partially display the card number.
msisdn	MSISDN	MSISDN number used to identify an alias in instant payments.
currency	Currency code	The account currency code.

4. Account Details

Attribute	Type	Description
resourceId	String	Data element used to access data from a dedicated account.
iban	IBAN	IBAN reference of the account.
bban	BBAN	BBAN code, used for accounts without an IBAN.
pan	PAN	Account number for cards, used instead of IBAN or BBAN.
maskedPan	Masked PAN	Masked PAN for partially displaying the card number.
msisdn	MSISDN	MSISDN number used to identify an alias in instant payments.
currency	Currency code	The account currency code (ISO 4217).
ownerName	Max140Text	The legal owner's name of the account.
name	Max70Text	The account name, assigned by the ASPSP in collaboration with the account holder for additional identification.
displayName	Max70Text	The account name as defined by the PSU in online channels.
product	Max35Text	The product name associated with this account.
cashAccountType	Cash Account Type	The account type code according to ISO 20022.
status	String	The account status ("enabled" – available, "deleted" – closed, "blocked" – blocked for legal reasons).
bic	BICFI	The BIC code associated with the account.

Attribute	Type	Description
linkedAccounts	Max70Text	Name for the account associated with pending card transactions.
usage	Max4Text	Specifies the usage of the account (PRIV – personal, ORGA – professional).
details	Max500Text	Additional details about the account or card features.
creditLimit	Amount	The credit limit of the PSU for all cards linked to this card account.
balances	Array of Balances	Specifies the account balances.
_links	Links	Direct links to access detailed information about the account, balances, or transactions.

5. Links

Attribute	Description
scaRedirect	Link to the ASPSP where SCA is performed using the Redirect SCA approach.
scaOAuth	The link refers to a JSON document specifying the OAuth details of the ASPSP's authorization server.
confirmation	The link that defines the URL to a resource that must be updated with a confirmation code or an access token, depending on the authentication process (Redirect or OAuth).
startAuthorisation	Link to an endpoint where transaction authorization or cancellation starts with a POST. No additional data required.
startAuthorisationWithPsuIdentification	Link to an endpoint where transaction authorization starts with the PSU identification.
updatePsuIdentification	Link to the payment initiation or account information resource, which requires an update with the

Attribute	Description
	PSU identification if it hasn’t been provided already.
startAuthorisationWithProprietaryData	Link to an endpoint where transaction authorization starts with specific user data according to the ASPSP documentation.
updateProprietaryData	Link to the resource that requires an update with the provided data.
startAuthorisationWithPsuAuthentication	Link to an endpoint where authorization starts with PSU authentication.
updatePsuAuthentication	Link to the resource that requires an update with the PSU password and, possibly, PSU identification if it hasn't been provided yet.
updateAdditionalPsuAuthentication	Link to the resource that requires an update with an additional PSU password.
startAuthorisationWithEncryptedPsuAuthentication	Link to an endpoint where authorization starts with providing the PSU's encrypted authentication data.
updateEncryptedPsuAuthentication	Link to the resource that requires an update with the PSU’s encrypted password and, possibly, PSU identification.
updateAdditionalEncryptedPsuAuthentication	Link to the resource that requires an update with an additional encrypted PSU password.
startAuthorisationWithAuthenticationMethodSelection	Link to an endpoint where authorization starts with selecting the SCA method.
selectAuthenticationMethod	Link to a resource where the TPP can select the applicable authentication method for the PSU.

Attribute	Description
startAuthorisationWithTransactionAuthorisation	Link to an endpoint where transaction authorization and the transmission of response data for the challenge are done simultaneously
authoriseTransaction	Link to the payment initiation or consent resource where the transaction authorization request is sent.
self	Link to the payment initiation resource created via request. Can be used to check the transaction status.
status	Link to check the status of the transaction resource.
scaStatus	Link to check the status of the authorization or cancellation of the authorization.
account	Link to the resource that provides details about a specific account.
balances	Link to the resource that provides the balance of a specific account.
transactions	Link to the resource that provides the transaction history of a specific account.
cardAccount	Link to the resource that provides details about a card account.
cardTransactions	Link to the resource that provides the transaction history of a card account.
transactionDetails	Link to the resource that provides details about a specific transaction.
first	Navigation link for paginated account reports.

Attribute	Description
next	Navigation link for the next paginated report.
previous	Navigation link for the previous report.
last	Navigation link for the last paginated report.
download	Download link for large AIS data packages.

6. SCA Status

Code	Description
received	The authorization resource has been successfully created.
psuIdentified	The PSU has been identified for the authorization resource.
psuAuthenticated	The PSU has been identified and authenticated.
scaMethodSelected	The SCA procedure has been selected by the PSU/TPP.
started	The SCA procedure has started.
unconfirmed	SCA has been technically completed successfully, but confirmation is required from the TPP.
finalised	The SCA procedure has been successfully completed.
failed	The SCA procedure has failed.

7. Transaction Status

Code	Description
ACCC	The settlement on the creditor's account has been completed.
ACCP	The customer profile verification was successful.
ACSC	The settlement on the debtor's account has been completed.
ACSP	The payment initiation has been accepted for execution.

Code	Description
ACTC	The technical validation has been accepted.
ACWC	The instruction has been accepted, but a modification will be made.
RCVD	The payment initiation has been received by the receiving agent.
PDNG	The payment initiation or transaction is pending.
RJCT	The payment initiation or transaction has been rejected by the PSU on the ASPSP side.
CANC	The payment initiation has been cancelled before execution upon a request from the TPP via the API.

8. Balance Type

Balance Type	Description
closingBooked	The account balance at the end of the agreed reporting period. It includes all entries made on the account.
expected	Balance calculated based on completed entries and pending items, projecting the balance at the end of the day.
openingBooked	The account balance at the start of the reporting period, identical to the final balance of the previous report.
interimAvailable	The available balance calculated throughout the business day, subject to subsequent changes during the day.
interimBooked	The balance calculated during the day, based on the credit and debit entries made up to the specified time.
forwardAvailable	The available balance in advance, which the account holder can use on a specified date.
nonInvoiced	For card accounts only, currently under definition.

9. HTTP Response

Status Code	Description
200 OK	Response code for PUT, GET requests, indicating the request was successful.
201 Created	POST response for a correctly executed Payment Initiation or Consent request.

Status Code	Description
202 Accepted	Response code for the DELETE method, used when a payment can be cancelled but requires additional authorization.
204 No Content	Response code for the DELETE method, indicating the consent resource has been successfully deleted/revoked. The request has been processed, but the response contains no content (no content).
400 Bad Request	Validation error, e.g., incorrect syntax in the request or incorrect data in the payload.
401 Unauthorized	The TPP or PSU is not properly authorized to make the request.
403 Forbidden	The resource exists but cannot be accessed by the TPP or PSU.
404 Not Found	The resource or endpoint referred to does not exist or cannot be found.
405 Method Not Allowed	The HTTP method is not supported on a specific endpoint.
408 Request Timeout	The server is functioning correctly, but the request has timed out.
409 Conflict	The request was not completed due to a conflict with the current state of the target resource.
415 Unsupported Media Type	The TPP provided a media type (Content-Type) that the ASPSP does not support.
429 Too Many Requests	The request was rejected due to too many requests in a short period of time.
500 Internal Server Error	General server error from the ASPSP, indicating an internal issue.
503 Service Unavailable	The service is temporarily unavailable, usually due to maintenance or server overload.

10. HTTP Error Codes

Code	HTTP code	Description	Endpoint method
FORMAT_ERROR	400	The format of certain fields in the request does not meet the requirements.	/consents, /accounts, /payments

PARAMETER_NOT_CONSISTENT	400	The parameters sent by the TPP are inconsistent (only for query parameters).	/consents, /accounts, /payments
PARAMETER_NOT_SUPPORTED	400	The parameter is not accepted by the ASPSP API.	/consents, /accounts
SERVICE_INVALID	400 (if payload)/ 405 (if HTTP method)	The requested service is not valid for the requested resources.	/consents, /accounts, /payments
RESOURCE_UNKNOWN	400 (if payload)/ 403 (if another resource in the path)/ 404 (if account-id in the path)	The requested resource cannot be found in relation to the TPP.	/consents, /accounts, /payments
RESOURCE_EXPIRED	400 (if payload)/ 403 (if path)	The requested resource has expired and is no longer accessible.	/consents, /accounts, /payments
RESOURCE_BLOCKED	400	The requested resource cannot be accessed, as it is blocked, for example, in a signing basket.	/consents, /accounts, /payments
TIMESTAMP_INVALID	400	The time is outside an accepted period.	/consents, /accounts, /payments
PERIOD_INVALID	400	The requested time period is out of range.	/consents, /accounts, /payments
SCA_METHOD_UNKNOWN	400	The SCA method approached in the authentication method selection request is unknown or cannot be matched by the ASPSP with the PSU.	/consents, /accounts, /payments
CONSENT_UNKNOWN	400 (if header)/403 (if path)	The Consent-ID cannot be found by the ASPSP in relation to the TPP.	/consents, /accounts, /payments
SESSIONS_NOT_SUPPORTED	400	The indicator for the combined service cannot be used with this ASPSP	/consents, /accounts, /payments
PAYMENT_FAILED	400	The POST request for payment initiation failed. The ASPSP will provide the failure details.	/payments
EXECUTION_DATE_INVALID	400	The requested execution date is invalid for the ASPSP.	/payments
CERTIFICATE_INVALID	401	The content of the signature certificate does not meet the requirements or is invalid.	/consents, /accounts, /payments

ROLE_INVALID	403	The TPP does not have the required role.	/consents, /accounts
CERTIFICATE_EXPIRED	401	The signature certificate has expired.	/consents, /accounts, /payments
CERTIFICATE_BLOCKED	401	The signature certificate has been blocked.	/consents, /accounts, /payments
CERTIFICATE_REVOKED	401	The signature certificate has been revoked.	/consents, /accounts, /payments
CERTIFICATE_MISSING	401	The signature certificate was not provided in the request but is required for the corresponding service.	/consents, /accounts, /payments
CERTIFICATE_UNKNOWN	401	The signature certificate is not found in the Open Banking Digital List.	/consents, /accounts, /payments
SIGNATURE_INVALID	401	The signature applied for TPP authentication is invalid.	/consents, /accounts, /payments
SIGNATURE_MISSING	401	The signature applied for TPP authentication is required but is missing.	/consents, /accounts, /payments
CORPORATE_ID_INVALID	401	PSU-Corporate-ID cannot be found by the ASPSP.	/consents, /accounts, /payments
PSU_CREDENTIALS_INVALID	401	PSU-ID cannot be found by the ASPSP, is blocked, or the password/OTP is incorrect.	/consents, /accounts, /payments
CONSENT_INVALID	401	The consent created by the TPP is not valid for the requested service/resource.	/consents, /accounts, /payments
CONSENT_EXPIRED	401	The consent created by the TPP has expired and needs to be renewed.	/consents, /accounts, /payments
TOKEN_UNKNOWN	401	The OAuth2 token cannot be found by the ASPSP in relation to the TPP.	/consents, /accounts, /payments
TOKEN_INVALID	401	The OAuth2 token is associated with the TPP but is not valid for the requested service/resource.	/consents, /accounts, /payments
TOKEN_EXPIRED	401	The OAuth2 token has expired and needs to be renewed.	/consents, /accounts, /payments

SERVICE_BLOCKED	403	The service is not accessible to the PSU due to a channel-independent block by the ASPSP.	/consents, /accounts, /payments
PRODUCT_INVALID	403	The requested payment product is not available for the PSU.	/payments
PRODUCT_UNKNOWN	404	The requested payment product is not supported by the ASPSP.	/payments
CANCELLATION_INVALID	405	The targeted payments cannot be canceled due to a time limit or legal restrictions.	/payments
REQUESTED_FORMATS_INVALID	406	The requested resource does not allow additional authorizations.	/consents, /accounts
STATUS_INVALID	409	The requested resource does not allow additional authorizations.	/consents, /accounts, /payments
ACCESS_EXCEEDED	429	Access to the account has exceeded the consented multiplicity without PSU involvement per day.	/consents, /accounts

The mechanism for ASPSP verification of the requests transmitted by the TPP

1. Generation of the request by the TPP

Step 1. Calculating the hash for the Digest field

1.1. The TPP creates the body of the request (e.g., JSON with transaction or consent data).

1.2. The TPP applies the SHA-256 algorithm to the request body to generate a unique hash.

1.3. The resulting hash is encoded in Base64 format and placed in the Digest header of the request.

Step 2. Creating the signing string

2.1. The TPP prepares a signing string based on specific fields from the request headers.

2.2. The signing string is obtained by concatenating the values of the signed headers in a specific format.

2.3. The order of the header values is important and follows the specifications.

Step 3. Signing the signing string

3.1. The TPP uses its private RSA key associated with its certificate to generate the electronic signature.

3.2. The signature is applied to the signing string using the RSA-SHA256 algorithm.

3.3. The resulting electronic signature is encoded in Base64 and placed in the Signature header.

Step 4. Adding the certificate to the TPP-Signature-Certificate header

4.1. The TPP includes its public certificate in the TPP-Signature-Certificate header of the request.

4.2. The certificate is issued by a trusted Certification Authority (CA) and contains the TPP's public key.

Step 5. Sending the request to the ASPSP

5.1. The TPP sends the complete request, including the Digest, Signature, and TPP-Signature-Certificate headers, to the ASPSP endpoint.

2. Request verification by the ASPSP

Step 1. Validation of the TPP certificate

1.1. The ASPSP verifies the TPP certificate included in the TPP-Signature-Certificate header.

Steps:

1.1.1. Verify the certificate's trust chain up to a trusted Certification Authority (CA).

1.1.2. Verify that the certificate is valid (not expired and not revoked).

1.1.3. The ASPSP must verify that the certificate presented by the TPP is issued and valid for use in Open Banking services, specifically for electronic signing and authentication, and matches exactly the TPP registered or licensed by the NBM (by querying the NBM Open Banking Digital List if used by ASPSP for TPP verification), by validating the certificate's serial number and issuing authority (level 2 CA or root CA), as well as the Signature field.

Result: If the certificate is valid, the TPP's identity is confirmed.

Step 2. Verifying the Hash in the Digest

2.1. The ASPSP calculates its own SHA-256 hash of the received request body.

2.2. The calculated hash is compared with the value in the Digest header.

Result: If the two hashes match, the integrity of the request body is confirmed.

Step 3. Verifying the signature in the Signature Header

3.1. The ASPSP extracts the header values included in the "signing string" based on the information in the Signature.

3.2. The ASPSP recreates the signing string following the same format used by the TPP.

3.3. The ASPSP uses the public key extracted from the certificate in the TPP-Signature-Certificate header to verify the signature in the Signature.

Result: If the signature is valid, the authenticity of the request is confirmed.

Step 4. Authorizing the request

4.1. If all verifications pass (role, integrity, signature, and certificate), the ASPSP processes the request.

4.2. Otherwise, the request is rejected with an error message (e.g., 401 Unauthorized or 403 Forbidden).

3. Additional Details

How does the ASPSP verify the signature in the Signature header?

1. Extract the public key from the TPP-Signature-Certificate:
   1.1. The ASPSP retrieves the certificate (TPP-Signature-Certificate) from the TPP-Signature-Certificate header of the request.
   1.2. From this certificate, the ASPSP extracts the public key of the TPP.

2. Recreate the signing string:
    2.1. The ASPSP analyzes the Signature header to identify the signed fields (e.g., Digest, X-Request-ID, Date, etc.).
    2.2. The ASPSP recreates the signing string from the actual values of these headers in the specified order.

3. Decode the signature:
    3.1. The ASPSP decodes the value from the signature field in the HTTP Signature header (i.e., Signature.signature), which represents the electronic signature generated by the TPP. The decoding is done using the public key extracted from the TPP-Signature-Certificate and the algorithm specified in Signature.algorithm (e.g., RSA-SHA256).

4. Compare the resulting hash:
    4.1. During decoding, the ASPSP obtains a hash that was calculated by the TPP at the time of signing.
    4.2. The ASPSP compares this hash with the hash generated internally from the recreated signing string.

5. Possible results:
    5.1. If the hashes match, the signature is valid.
    5.2. If the hashes do not match, the signature is invalid, and the request is rejected.