Partner
in Payments

User Guide

SIBS API Market

PSD2 APIs

Third party payment service providers (TPP)

Version: 01.51

Date: 2026-01-06

Status: Final

Classification: Restricted

Reference: DCSIBS230100

Document Info

Reference:	DCSIBS230100
Document Title:	SIBS API Market - PSD2 APIs
Version:	01.51
Status:	Final
Classification:	Restricted
Document Type:	User Guide

Related Documents

Reference	Title	Source
N/A	N/A	N/A

Version History

Version	Date	Description	Author
01.04	2019-02-25	First version for public release	SIBS FPS
01.05	2019-03-12	Changes to this version: • Table in section 2.10 ASPSP's specific options updated • API endpoints' address included in section 2.2 API endpoints' address structure updated	SIBS FPS
01.06	2019-04-10	Changes to this version: • New chapter "message signing" included • Adjustments made on chapters 2.2 API endpoints' address structure and 2.3 Character Set.	SIBS FPS
01.07	2019-08-20	Changes to this version: • Character set has been updated in section 2.3 • Section 2.7 renamed and updated • New Section 2.9 • Section 2.10 updated • Clarification on Digest computation on section 4.2 • Date format of the example message amended on section 4.4	SIBS FPS
01.08	2019-12-12	Changes to this version: • Character set has been updated • Included information on MULTIBANCO Payment APIs • Section 2.9 and 2.10 were updated	SIBS FPS
01.10	2020-11-25	Changes to this version: • New sections 2.4 - reference to Message Codes, 2.11 App-to-app redirection and 2.12, Account Information API - Interpretation of Balances Fields for Card Accounts were included • New chapter 3 Developers Portal Functionalities was included • Section 2.9 was updated • Section 2.10 was updated • Section 6.2 was updated	SIBS FPS
01.20	2021-03-23	Changes to this version: • Section 2.9 was updated	SIBS FPS
01.21	2021-05-27	Changes to this version: • Section 2.9 was updated	SIBS FPS
01.22	2021-12-17	Changes to this version: • Section 2.9 was updated • Section 6.1 was updated • Section 2.13 was created	SIBS FPS

Version	Date	Description	Author
01.23	2023-03-31	Changes to this version: • Section 2 was updated, namely the subsections 2.2 and 2.10 • Section 6.1 and its subsections were updated • Section 6.2 and its subsections were updated • Section 6.3 was created • Section 7 was created	SIBS FPS
01.30	2023-09-14	Changes to this version: • Section 6.2 was updated; • Section 7 was updated.	SIBS FPS
01.40	2024-03-21	Changes to this version: • Table 1 of section 2.7 was updated; • Table 3 of section 2.10 was updated; • Section 7 was updated.	SIBS FPS
01.50	2025-11-24	Changes to this version: • Table 3 of section 2.10 was updated; • Section 4.3 was updated; • Section 6.1.1 was updated; • Section 6.2 was updated, including changes to table 5; • FAQs section (section 7) was updated. • Other edit changes, throughout the document, without any impact on the technical information.	Service Development
01.51	2026-01-06	Changes to this version: • Section 4 Message signing was updated; • Section 7 FAQs was updated.	Service Development

Table of Contents

List of Figures

List of Tables

1 Introduction

PSD2 APIs provided in SIBS API Market allow Account Information Service Providers (AISP), Payment Initiation Service Providers (PISPs) and Payment Service Providers issuing Card Based Payment Instruments (CBPIIs), collectively known as Third Party Payment Service Providers or TPPs, which are authorised by a National Competent Authority under [PSD2] scope, to access payment accounts on Account Servicing Payment Service Providers (ASPSPs) that have selected SIBS API Market to open their accounts to these new players.

The APIs under [PSD2] scope are:

- Account Information – can be used by AISPs;
- Payment Initiation – can be used by PISPs;
- Funds Confirmation – can be used by CBPIIs and PISPs.

With just one integration, SIBS API Market allows TPPs to reach all ASPSPs listed in section 2, through a common set of APIs, covering more than 95 % of the payment accounts held by PSUs in Portugal. The list of ASPSPs can also be retrieved through the List of Banks API available on SIBS API Market. The intended ASPSP is selected by the TPP, on each API call, in a parameter that is part of the API endpoint path.

1.1 Objective

This document aims to provide information on the usage of PSD2 APIs available on SIBS API Market to TPPs.

1.2 Scope

This document covers the implementation choices made by ASPSPs on SIBS API Market in relation to the reference specifications, and provides guidance on the sequence in which each API's operation shall be executed.

The detailed specification of the APIs and API parameters is beyond the scope of this document.

1.3 References

Reference	Title
[BG-IG]	The Berlin Group Joint Initiative on a PSD2 Compliant XS2A Interface; NextGenPSD2 XS2A Framework; Implementation Guidelines; Version 1.3.12; 01 July 2022.
[eIDAS]	REGULATION (EU) No 910/2014 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC.
[ETSI-PSD2]	TECHNICAL SPECIFICATION ETSI TS 119 495 - Electronic Signatures and Infrastructures (ESI); Sector Specific Requirements; Qualified Certificate Profiles and TSP Policy Requirements under the payment services Directive (EU) 2015/2366.

Reference	Title
[PSD2]	DIRECTIVE (EU) 2015/2366 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 25 November 2015 on payment services in the internal market.
[RTS]	COMMISSION DELEGATED REGULATION (EU) 2018/389 of 27 November 2017 supplementing Directive (EU) 2015/2366 of the European Parliament and of the Council regarding regulatory technical standards for Strong Customer Authentication and common and secure open standards of communication.

1.4 Definitions

For the purposes of this document, the following definitions apply:

AISP Account Information Service Provider as defined in [PSD2].

ASPSP Account Servicing Payment Service Provider as defined in [PSD2] (e.g., Banks).

CBPII Card Based Payment Instrument Issuer as defined in [PSD2].

NCA Nacional Competent Authority as defined in [PSD2].

PISP Payment Initiation Service Provider as defined in [PSD2].

PSU Payment Services User as defined in [PSD2].

QTSP Qualified Trust Service Provider as defined in [eIDAS].

SCA Strong Customer Authentication as defined in [RTS].

TPP Third Party Payment Services Provider authorised by a National Competent Authority to provide payment services according to one or more of the following roles defined in [PSD2]:

- Payment Initiation Service Provider (PISP);
- Account Information Service Provider (AISP);
- Card-Based Payment Instrument Issuer (CBPII).

2 Implementation options

The specification of the PSD2 APIs provided by all ASPSPs on SIBS API Market is based on version 1.3.12. (July 2022) of NextGenPSD2 XS2A Framework of Berlin Group [BG-IG].

The NextGenPSD2 Guidelines leave it up to ASPSPs to decide whether to implement certain optional features. This section identifies the choices made in implementing SIBS API Market for PSD2 APIs.

2.1 Qualified certificates

For identification purposes when accessing PSD2 APIs on SIBS API Market, TPPs shall use a qualified certificate for electronic seals (QSealC) and a qualified certificate for website authentication (QWAC) with a PSD2 profile in accordance with [ETSI-PSD2], issued by a Qualified Trust Service Provider (QTSP) recognised by a competent authority under [eIDAS] regulation. All request messages directed to SIBS API Market API shall be signed with the private key associated with the public key included in the QSeal Certificate (see details in section 2.11). QWAC shall be used to establish a secure channel for communication between the TPP and SIBS API Market using the Transport Layer Security (TLS) protocol.

The above-mentioned certificates are not required for testing on SIBS API Market Sandbox environment.

After applying for authorisation from a national competent authority, and while the competent authority does not grant authorisation, TPPs may use test certificates in the SIBS API Market Test/Production environment, while in test mode.

2.2 API endpoint address structure

The API endpoint address follows the general structure defined in [BG-IG]:

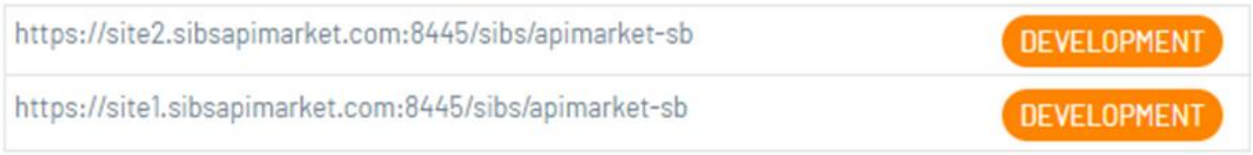
- https://{provider}/v1/{service}{?query-parameters}

where

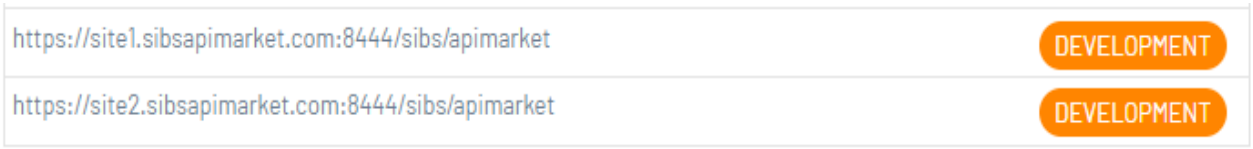
- {provider} consists of {host}/{path}/{aspsp-cde}

SIBS API Market infrastructure is divided into two active/active redundant sites. The {host}/{path} part of the API endpoint address defines the site and environment (development or production) that will process the API call. The values to use in the {host}/{path} part of the API endpoints are available in the API documentation for each API in the Sandbox and Test & Production environments. Sandbox environment only provides Development endpoints for site 1 and site 2. Test & Production environment provides Development and Production endpoints for site 1 and site 2.

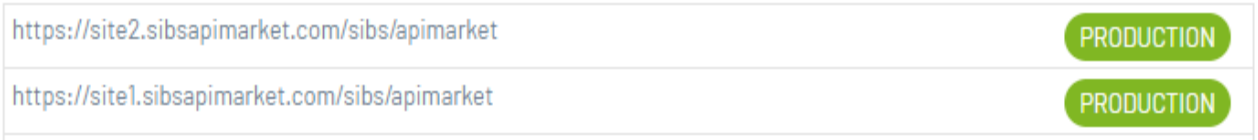
Example of the {host}/{path} part of the Payment Initiation API endpoints for testing in the SIBS API Market Sandbox environment:



Example of the {host}/{path} part of the Payment Initiation API endpoints for end-to-end testing with ASPSPs in the SIBS API Market Test & Production environment:



Example of the {host}/{path} part of the Payment Initiation API endpoints for production in the SIBS API Market Test & Production environment:



The {aspsp-cde} part of the API endpoint address defines the ASPSP that the TPP wants to call for the provision of the service requested in the {service} part of the API endpoint. The possible values for aspsp-cde can be obtained through the List of Banks API (e.g., BBPI, BNKI, BST and CCAML).

Example of calling an API in the Sandbox environment of the ASPSP CCAML via site 1:

- https://site1.sibsapimarket.com:8445/sibs/apimarket-sb/CCAML/...

2.3 Character set

SIBS API Market accepts the following character set in messages:

- a b c d e f g h i j k l m n o p q r s t u v w x y z
- A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
- 0 1 2 3 4 5 6 7 8 9
- / - ? : ( ) . , ' + < = > & % _ * #
- Space

Messages containing characters outside this set are declined, except for the parameter TPP-Certificate, where all characters are allowed.

2.4 Message codes

The list of message codes that can be returned by the APIs is published on the Developers Portal and is based on the Berlin Group implementation guidelines.

2.5 Account reference

Berlin Group's NextGen PSD2 framework allows for the use of several different types of references for payment accounts/cards (e.g.: iban, pan, msisdn). SIBS API Market implementation supports "iban" and "bban" for account references, and "pan" and "mpan" for card-account references.

2.6 Options not supported

The following features included in [BG-IG] are not supported by ASPSPs on SIBS API Market:

- OAuth2 protocol for PSU authentication;
- Balances in list of accounts (GET accounts?withBalance);
- Balances in account details (GET accounts/{account-id}?withBalance);
- Balances in list of transactions (GET accounts/{account-id}/transactions ?withBalance);
- Transaction details (GET accounts/{account-id}/transactions/{transaction-id});
- Delta access in list of transactions (GET accounts/{account-id}/transactions ?transactionId and GET accounts/{account-id}/transactions?deltaList).

2.7 Supported APIs

All APIs included in [BG-IG] are, or will be, supported in the Sandbox and Test & Production environments of SIBS API Market, in stages, according to the plan defined in Table 1.

APIs supported in the Sandbox environment are available in all ASPSPs.

APIs supported in the Test & Production environment may not be available in all ASPSPs, depending on their own implementation plans and whether the same operation is supported in their direct PSU interfaces (refer to section 2.10).

The APIs, as well as the API versions, available at each ASPSP can be retrieved using the List of Banks API, available in the Information APIs product of SIBS API Market.

The list of APIs available at each ASPSP provided by the List of Banks API may differ in the environment type and, for the Test and Production environment, in the Test and Production endpoint type, depending on the APIs that each ASPSP has implemented in each environment/endpoint.

The List of Banks API in the Sandbox environment returns the list of APIs, and API versions, available in each ASPSP in the Sandbox environment for closed-loop testing with static data.

The List of Banks API in the Test (DEVELOPMENT) endpoints of the Test & Production environment returns the list of APIs, API versions, payment-product (e.g.: SEPA CT) available in each ASPSP for end-to-end testing with the ASPSP using non-real PSU accounts/cards.

The List of Banks API in the Production (PRODUCTION) endpoints of the Test & Production environment returns the list of APIs, API versions, payment-product (e.g.: SEPA CT) available in each ASPSP for access to real PSU accounts/cards.

Payment products of the payments are also returned by the List of Banks API.

Table 1 - SIBS Market support for Berlin Group APIs

API	Operation	Sandbox	Test & Production
payments/{payment-product}	Initiation	2019Q1	2019Q1/Q2[1]
payments/{payment-product}	Cancellation	2019Q1	2019Q3[1]
periodic-payments/{payment-product}	Initiation	2019Q1	2019Q2[1]
periodic-payments/{payment-product}	Cancellation	2019Q1	2019Q3[1]
consents	All	2019Q1	2019Q1
accounts	All	2019Q1	2019Q1
funds-confirmations	For CBPII/PISP	2019Q1	2019Q1
bulk-payments/{payment-product}	All	2020Q1	2020Q1
cards accounts	All	2022Q3/Q4	2023Q4[1]
consent authorisation	All	2022Q3/Q4	2023Q4

[1] The availability of the API and payment products may differ between ASPSPs. Please refer to the information above on how to use the List of Banks API to find out which ASPSPs have implemented this API. Please refer to section 2.10 to find out which products are available at each ASPSP.

2.8 Payment product fallback in payment initiation APIs

Whenever both the Debtor and the Creditor of the payment account/card account are held by the same ASPSP providing the payment initiation service, the payment product used to perform the payment may, depending on the ASPSP, be changed to an internal credit transfer, with the funds immediately available to the Creditor, regardless of the value of the path parameter “payment-product”.

Whenever both the Debtor and the Creditor of the payment account/card account are held by ASPSPs belonging to the SEPA area and the currency is Euro, the payment product used to perform the payment may, depending on the ASPSP providing the payment initiation service, be changed to a SEPA CT payment product, when the path parameter “payment-product” is “cross-border-credit-transfers”.

In both cases, the rules applied to the transaction (including the cost to the Debtor and/or Creditor) are the same as for internal credit transfer and SEPA CT payments carried out on ASPSP channels (e.g.: home banking).

2.9 APIs for domestic payment products

In addition to [BG-IG] APIs, SIBS API Market provide payment initiation APIs, under PSD2, for payment products specific to the Portuguese market.

These APIs are supported in the SIBS API Market Sandbox and Test & Production environments, as shown in Table 2.

The multibanco-payment-type and service-payment-name path parameters are returned by the MULTIBANCO Payments Catalogue API (GET multibanco-payments/service-catalogue).

Table 2 - SIBS Market support for domestic payment APIs

API
multibanco-payments/service-catalogue
multibanco-payments/{multibanco-payment-type}/{service-payment-name}
periodic-multibanco-payments/{multibanco-payment-type}/{service-payment-name}
bulk-multibanco-payments/{multibanco-payment-type}
tsu-payments/{payment-product}
bulk-tsu-payments/{payment-product}

2.10 Features supported by each ASPSP

In the Sandbox environment, API data is static and implemented options are the same for all ASPSPs.

In the Test and Production environments, not all features available on SIBS API Market are provided by all ASPSPs, i.e. are available on their online interfaces for PSUs.

The following table shows the list of features supported in the Test and Production environments by each of the ASPSPs present on SIBS API Market for Private and Corporate payment accounts.

The List of Banks API provides you real-time information on the features available at each ASPSP and should always be used as a reference.

Table 3 - ASPSPs present on SIBS API Market - features supported on Test and Production environments

	Supported payment products[1] for single payment	Supported authentication approaches	Support for combined AI/PI access	Supported payment products[1] for future dated payments	Supported payment products[1] for periodic payments	Supported "executionRule" for periodic payments	Supported "frequency" for periodic payments	Support of "dayOfExecution"[2] for periodic payments	Supported payment products[1] for bulk payments	Support of App-to-app redirection	Support of Chargebearer as mandatory field
ActivoBank by Millennium	• SEPA CT • SCT Inst • TARGET • Cross Border • Pag. de Serviços • Pag. ao Estado • Carreg. Telemóveis • Pag. Seg. Soc. • Pag. TSU	• Redirect	Yes	• SEPA CT • SCT Inst	• SEPA CT • SCT Inst	• Preceding • Following	• Daily • Weekly • Monthly • Every two months • Quarterly • Semiannual • Annual	No	No	Yes	No
ATLANTICO BANCO ATLANTICO EUROPA	• SEPA CT • SCT Inst • TARGET • Cross Border • Pag. de Serviços • Pag. ao Estado • Carreg. Telemóveis • Pag. Seg. Soc. • Pag. TSU	• Redirect	Yes	• SEPA CT	• SEPA CT	• Following	• Weekly • Monthly • Quarterly • Semiannual • Annual	Yes	No	Yes	No

	Supported payment products[1] for single payment	Supported authentication approaches	Support for combined AI/PI access	Supported payment products[1] for future dated payments	Supported payment products[1] for periodic payments	Supported "executionRule" for periodic payments	Supported "frequency" for periodic payments	Support of "dayOfExecution"[2] for periodic payments	Supported payment products[1] for bulk payments	Support of App-to-app redirection	Support of Chargebearer as mandatory field
bankinter	• SEPA CT • SCT Inst • TARGET • Cross Border • Pag. de Serviços • Pag. ao Estado • Carreg. Telemóveis • Pag. Seg. Soc.	• Redirect	Not supported	• SEPA CT • Pag. de Serviços • Pag. ao Estado • Carreg. Telemóveis • Pag. Seg. Soc.	• SEPA CT • Pag. de Serviços • Pag. ao Estado • Carreg. Telemóveis • Pag. Seg. Soc.	• Preceding • Following	• Weekly • Every two weeks • Monthly • Every two months • Quarterly • Semiannual • Annual	Not supported	• SEPA CT	Yes	Not supported
BiG Banco de Investimento Global	• SEPA CT	• Redirect	Not supported	• SEPA CT	• SEPA CT	• Following	• Daily • Weekly • Monthly • Annual	Yes	Not supported	Not supported	Not supported
BiG Banco de Investimento Global Espanha	• SEPA CT	• Redirect	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported
BPI	• SEPA CT • SCT Inst • TARGET • Cross Border • Pag. de Serviços • Pag. ao Estado • Carreg. Telemóveis • Pag. Seg. Soc. • Pag. TSU	• Redirect	Not supported	• SEPA CT	• SEPA CT	• Following	• Daily • Weekly • Every two weeks • Monthly • Every two months • Quarterly • Semiannual • Annual	Yes	• SEPA CT • TARGET	Yes	Not supported
Banco Português de Gestão	• SEPA CT • SCT Inst	• Redirect	Yes	• SEPA CT	• SEPA CT	• Following	• Daily • Weekly • Every two weeks • Monthly • Every two months • Quarterly • Semiannual • Annual	Yes	Not supported	Not supported	Not supported

	Supported payment products[1] for single payment	Supported authentication approaches	Support for combined AI/PI access	Supported payment products[1] for future dated payments	Supported payment products[1] for periodic payments	Supported "executionRule" for periodic payments	Supported "frequency" for periodic payments	Support of "dayOfExecution"[2] for periodic payments	Supported payment products[1] for bulk payments	Support of App-to-app redirection	Support of Chargebearer as mandatory field
CA Crédito Agrícola	• SEPA CT • SCT Inst • Cross Border • Pag. de Serviços • Pag. ao Estado • Carreg. Telemóveis • Pag. Seg. Soc. • Pag. TSU	• Redirect	Yes	• SEPA CT • Pag. de Serviços	• SEPA CT	• Following	• Daily • Weekly • Every two weeks • Monthly • Quarterly • Semiannual • Annual	Yes	• SEPA CT • Pag. TSU	Yes	Not supported
Caixa de Crédito de Leiria	• SEPA CT • SCT Inst • Pag. de Serviços • Pag. ao Estado • Carreg. Telemóveis • Pag. Seg. Soc. • Pag. TSU	• Redirect	Not supported	• SEPA CT • Pag. de Serviços • Pag. ao Estado • Carreg. Telemóveis • Pag. Seg. Soc. • Pag. TSU	Not supported	Not supported	Not supported	Not supported	• SEPA CT • Pag. Seg. Soc. • Pag. TSU	Not supported	Not supported
Caixa Agrícola Bombarral	• SEPA CT • SCT Inst • Pag. Seg. Soc. • Pag. TSU	• Redirect	Yes	Not supported	Not supported	Not supported	Not supported	Yes	Not supported	Not supported	Not supported
Caixa de Crédito da Chamusca	• SEPA CT • SCT Inst • Pag. Seg. Soc. • Pag TSU	• Redirect	Yes	• SEPA CT	• SEPA CT	• Preceding • Following	• Daily • Monthly • Quarterly • Semiannual • annual	Yes	Pag. TSU	Not supported	Not supported
Caixa Agrícola de Mafra	• SEPA CT • SCT Inst • Pag. de Serviços • Pag. ao Estado • Carreg. Telemóveis • Pag. Seg. Soc. • Pag. TSU	• Redirect	Yes	Not supported	Not supported	Not supported	Not supported	Yes	Not supported	Not supported	Not supported

	Supported payment products[1] for single payment	Supported authentication approaches	Support for combined AI/PI access	Supported payment products[1] for future dated payments	Supported payment products[1] for periodic payments	Supported "executionRule" for periodic payments	Supported "frequency" for periodic payments	Support of "dayOfExecution"[2] for periodic payments	Supported payment products[1] for bulk payments	Support of App-to-app redirection	Support of Chargebearer as mandatory field
Caixa Agrícola de Torres Vedras	• SEPA CT • SCT Inst • Pag. Seg. Soc. • Pag. TSU	• Redirect	Yes	Not supported	Not supported	Not supported	Not supported	Yes	Not supported	Not supported	Not supported
CEM Caixa Económica da Misericórdia de Angra do Heroísmo	• SEPA CT • SCT Inst • Pag. de Serviços • Pag. ao Estado • Carreg. Telemóveis • Pag. Seg. Soc. • Pag. TSU	• Redirect	Not supported	• SEPA CT • Pag. de Serviços • Carreg. Telemóveis • Pag. TSU	• SEPA CT • Pag. de Serviços • Carreg. Telemóveis	• Following	• Daily • Weekly • Every two weeks • Monthly • Quarterly • Semiannual • Annual	Yes	• SEPA CT	Yes	Not supported
	• SEPA CT • SCT Inst • TARGET • Cross Border[3] • Pag. de Serviços • Pag. ao Estado • Carreg. Telemóveis • Pag. Seg. Soc. • Pag. TSU	• Redirect	Not supported	• SEPA CT • Cross Border • Pag. de Serviços • Pag. ao Estado • Carreg. Telemóveis	• SEPA CT • Cross Border • Pag. de Serviços • Pag. ao Estado • Carreg. Telemóveis	• Preceding	• Weekly • Every two weeks • Monthly • Every two months • Quarterly • Semiannual • Annual	Yes	• SEPA CT • Cross Border • Pag. de Serviços • Carreg. Telemóveis	Yes	Mandatory
Caixa Geral de Depositos France	• SEPA CT • SCT Inst	• Redirect	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported
Cofidis De pessoas para pessoas	• SEPA CT • SCT Inst	• Redirect	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported

	Supported payment products[1] for single payment	Supported authentication approaches	Support for combined AI/PI access	Supported payment products[1] for future dated payments	Supported payment products[1] for periodic payments	Supported "executionRule" for periodic payments	Supported "frequency" for periodic payments	Support of "dayOfExecution"[2] for periodic payments	Supported payment products[1] for bulk payments	Support of App-to-app redirection	Support of Chargebearer as mandatory field
Millennium bcp	• SEPA CT • SCT Inst • TARGET • Cross Border • Pag. de Serviços • Pag. ao Estado • Carreg. Telemóveis • Pag. Seg. Soc. • Pag. TSU	• Redirect	Yes	• SEPA CT • SCT Inst • Cross Border • Pag. de Serviços • Pag. ao Estado • Carreg. Telemóveis	• SEPA CT • SCT Inst • Cross Border • Pag. de Serviços • Pag. ao Estado • Carreg. Telemóveis	• Preceding • Following	• Weekly • Monthly • Every two months • Quarterly • Semiannual • Annual	Not supported	• SEPA CT • SCT Inst • TARGET • Cross Border • Pag. de Serviços • Pag. ao Estado • Carreg. Telemóveis • Pag. TSU	Yes	Not supported
Banco Montepio	• SEPA CT • SCT Inst • TARGET • Cross Border • Pag. de Serviços • Pag. ao Estado • Carreg. Telemóveis • Pag. Seg. Soc. • Pag. TSU	• Redirect	Not supported	• SEPA CT • Pag. de Serviços • Pag. ao Estado • Pag. TSU	• SEPA CT	• Following	• Daily • Weekly • Monthly • Every two months • Quarterly • Semiannual • Annual	Yes	• SEPA CT • SCT Inst	Yes	Not supported
novobanco	• SEPA CT • SCT Inst • Cross Border • Pag. de Serviços • Pag. ao Estado • Carreg. Telemóveis • Pag. Seg. Soc. • Pag. TSU	• Redirect	Not supported	• SEPA CT • SCT Inst • Cross Border • Pag. de Serviços • Carreg. Telemóveis • Pag. Seg. Soc.	• SEPA CT	• Following	• Weekly • Every two weeks • Monthly • Every two months • Quarterly • Semiannual • Annual	Yes	• SEPA CT	Yes	Not supported
novobanco DOS AÇORES	• SEPA CT • SCT Inst • Pag. de Serviços • Pag. ao Estado • Carreg. Telemóveis • Pag. Seg. Soc. • Pag. TSU	• Redirect	Not supported	• SEPA CT • SCT Inst • Pag. de Serviços • Carreg. Telemóveis • Pag. Seg. Soc.	• SEPA CT	• Following	• Weekly • Every two weeks • Monthly • Every two months • Quarterly • Semiannual • Annual	Yes	• SEPA CT	Yes	Not supported

	Supported payment products[1] for single payment	Supported authentication approaches	Support for combined AI/PI access	Supported payment products[1] for future dated payments	Supported payment products[1] for periodic payments	Supported "executionRule" for periodic payments	Supported "frequency" for periodic payments	Support of "dayOfExecution"[2] for periodic payments	Supported payment products[1] for bulk payments	Support of App-to-app redirection	Support of Chargebearer as mandatory field
Santander	• SEPA CT • SCT Inst • TARGET • Cross Border • Pag. de Serviços • Pag. ao Estado • Carreg. Telemóveis • Pag. Seg. Soc. • Pag. TSU	• Redirect	Not supported	• SEPA CT • SCT Inst • Pag. de Serviços • Pag. ao Estado • Carreg. Telemóveis • Pag. Seg. Soc. • Pag. TSU	• SEPA CT • SCT Inst	• Following	• Weekly • Monthly • Every two months • Quarterly • Semiannual • Annual	Not supported	• SEPA CT • SEPA CT Urg. • Cross Border • Pag. de Serviços • Pag. ao Estado	Yes	Cross Border Single Payment: Not mandatory. If empty, Santander assumes SHA (shared) by default. Cross Border Bulk Payment: Mandatory.
Unicre	• Pag. de Serviços • Pag. ao Estado • Carreg. Telemóveis	• Redirect	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported	Not supported

Note 1: SEPA CT: sepa-credit-transfers; Cross Border: cross-border-credit-transfers; SCT Inst: instant-sepa-credit-transfers; TARGET: target-2-payments.

Note 2: The date of a payment can be adjusted in accordance with the execution rule, the same rule that the ASPSP uses on his own channels (e.g., homebanking).

Note 3: At CGD, for cross-border payments, it is mandatory to fill in the debtor and creditor address fields, in accordance with ISO20022 usage guidelines.

2.11 App-to-app redirection

ASPSPs that offer an app to their users will make app-to-app redirection available to TPPs and support the authentication methods available on their own channels (e.g.: biometrics).

ASPSPs present on SIBS API Market that have already implemented app-to-app redirection support Universal Links in their iOS apps (https://developer.apple.com/ios/universal-links/) and App Links in their Android apps (https://developer.android.com/training/app-links) to handle app-to-app redirection for PSU authentication. Please refer to column "Support of App-to-app redirection" of the table included in section 2.10 to find out which ASPSPs already support app-to-app redirection.

2.11.1 Activating the ASPSP app

To activate the ASPSP app for PSU authenticate, TPPs shall request the Operating System to open the link returned by the ASPSPs using Universal Links and App Links mechanisms.

TPPs currently using Web Views to open the links returned by ASPSPs shall switch to the aforementioned mechanisms when opening links returned by "app-to-app ready" ASPSPs in order to benefit from app-to-app redirection.

The following examples show how to open links returned by ASPSPs depending on the Operating System (the example URLs must be replaced by the URLs returned).

iOS example (available in https://developer.apple.com/documentation/uikit/inter-process_communication/allowing_apps_and_websites_to_link_to_your_content)

```
if let appURL = URL(string:
"https://myphotoapp.example.com/albums?albumname=vacation&index=1") {
    UIApplication.shared.open(appURL) { success in
        if success {
            print("The URL was delivered successfully.")
        } else {
            print("The URL failed to open.")
        }
    }
} else {
    print("Invalid URL specified.")
}
```

Android example

```
Intent intent = new Intent (Intent.ACTION_VIEW);
intent.setData(Uri.parse("https://myphotoapp.example.com/albums?albumname=vacation&index=1"));
startActivity (intent);
```

2.11.2 Returning to the TPP app

Once the PSU is authenticated, the ASPSP app opens the callback URL sent by the TPPs in the API call (e.g., POST https://site1.sibsapimarket.com/sibs/apimarket/{aspsp-cde}/{v}/payments/sepa-credit-transfers), using the same mechanism with which the TPP opens the ASPSP app, as described in section 2.11.1.

TPPs must associate their callback URL with their app, as well as implement the handler in the app for receiving the redirect back, as described in https://developer.apple.com/ios/universal-links/ for iOS and https://developer.android.com/training/app-links for Android. This way, by the time the PSU is authenticated, the ASPSP redirects them back to the TPP app, which is associated with the TPP URL previously sent by the TPP.

2.12 Account Information API - Interpretation of balance fields for card accounts

When receiving information on a card account through the Account Information API, the TPP must interpret the fields relating to balances as follows:

- **interimAvailable** – Current available balance, calculated during the account servicer's business day at the specified time and subject to further changes during the business day. The interim balance is calculated based on booked credit and debit items during the specified calculation time/period;
- **authorised** – Credit limit, calculated by adding the expected[1] balance to the value of a pre-approved credit line the ASPSP makes permanently available to the user;
- **closingBooked** – Statement balance; the account balance at the end of the pre-agreed account reporting period. It is the sum of the opening booked[2] balance at the beginning of the period with all booked transactions entered on the account during the pre-agreed account reporting period.

[1] Expected – Balance composed of booked entries and pending items known at the time of calculation, which projects the end-of-day balance if everything is booked on the account and no other entry is posted.

[2] Opening booked – Book balance of the account at the beginning of the account reporting period. It is always equal the closing book balance of the previous report.

2.13 Account Information API – Navigation fields

TPPs shall not perform any validation or processing to the content of the parameters ‘first’, ‘previous’, ‘next’ and ‘last’ returned by SIBS API Market for navigation on the transaction pages returned by the GET transactions API. Navigation links are valid for 30 minutes and shall be used without changes to access the transaction pages on further requests.

The parameters used in these links are for SIBS internal use and may be changed at any time without notice.

3 Developers Portal functionalities

3.1 Support tickets

SIBS API Market website includes a page where any TPP can report an issue or question about the available APIs. The issues reported are tracked throughout each iteration between SIBS, the TPP who opened the ticket and the ASPSP for which the ticket was opened, until its successful resolution and closure.

Link: https://developer.sibsapimarket.com/live/support (for registered users only)

3.2 Developers Portal Forum

In this Developers Forum, TPPs can check scheduled downtimes by ASPSPs or discuss relevant features and uses of the APIs supported by the platform. By subscribing to the notifications feature, TPPs will be notified of all relevant information concerning the availability (e.g., scheduled maintenance) of the platform or an individual ASPSP.

Link: https://developer.sibsapimarket.com/live/forum

4 Message signing

All messages sent to SIBS API Market for PSD2 APIs in the Test & Production environment must be signed with the TPP private key associated with the public key included in the QSeal eIDAS certificate issued by an eIDAS QTSP.

All messages shall include the “TPP-Certificate”, “Digest” and “Signature” parameters.

4.1 TPP-Signature-Certificate

TPP-Signature-Certificate parameter must contain the QSeal eIDAS certificate issued by an eIDAS QTSP.

4.2 Digest

The “Digest” Header contains a message body Hash in the following format:

digest-algorithm=<encoded digest output>

Where:

digest-algorithm is the identifier of the algorithm used to compute the message hash. Possible values are SHA-256 and SHA-512.

<encoded digest output> is the base64 encoding of the result of the hash algorithm computed over the message body.

The message body must be in linear string format for the computation of the Hash. If needed, a conversion function shall be used (e.g., JSON.stringify()).

Example:

Message body	{
	"Hello": "World"
	}

Message body string	{"Hello":"World"}
Digest Header	Digest: SHA-256= ZaTEylrPxL87NOE4yAhzD2yc4UGkxriJqTReZslznXM=

4.3 Signature

The Signature parameter must contain the message signature in the following format (as per section 2 of “Signing HTTP Messages draft-cavage-http-signatures-12”, https://datatracker.ietf.org/doc/draft-cavage-http-signatures/):

keyId=”<key-identifier>”,algorithm=”<signature-algorithm>”,headers=”<header1> <header2> <headerN>”,signature=”<message-signature>”

Where:

<key-identifier> is the serial number of the TPP's Certificate included in the TPP-Signature-Certificate[3] parameter.

<signature-algorithm> is the identifier of the algorithm used to sign the message. Possible values are rsa-sha256 and rsa-sha512.

<header1>…<headerN> is the list of message header parameters included in the signing-string. The following message header parameters are mandatory and must be included: Digest, X-request-ID[4] and Date. The following message header parameters must be conditionally included:

- psu-id, if, and only if, PSU-ID is included as the HTTP-Request header;
- psu-corporate-id, if, and only if, PSU-Corporate-ID is included as the HTTP-Request header;
- ~~tpp-redirect-uri, if, and only if, TPP-Redirect-URI is included as the HTTP-Request header.~~

No further entries should be included.

<message-signature> is the base64 encoding of the signature algorithm result computed over the signing-string, using the private key that is the pair of the public key included in the TPP's Certificate sent in the TPP-Signature-Certificate[3] parameter.

The signing-string shall be assembled as the concatenation of the parameter names and values identified in the "headers", according to the following rules:

- All HTTP header names are in lowercase;
- All HTTP header names are immediately followed by an ASCII colon ':' (no ASCII spaces (chr(20)) in between);
- A single ASCII space (chr(20)) is added between the colon ':' and the header parameter value;
- All header parameter values are trimmed (leading and trailing ASCII spaces (chr (20)) are removed);
- If the header is not the last, then append an ASCII newline '\n' (one character (chr(0A))) just after the header parameter value.

Example:

```
Signature: keyId="44ba0c31580926fcda18ddee8d1ac0a9",algorithm="rsa-
sha256",headers="Digest ~~TPP-Transaction-ID TPP-Request-ID~~ X-Request-ID PSU-ID
Date",signature="H+5o5vQJ1KsSlfTG5hCehXxk63warUpkcBftyAuRsZEP2KCjraZZU0yu4IRSA6txo6FhLXB
HQND3e4VfSzvvi8pxdWIxqX8/lOo4EKiVz1jZkzyO5T1hcA+eInBKNxfb5vvk6wmfJ2FxoBJ9ba8JqH1txzjXhuP
Gj3j+Bcc9PTGHtg+U5Z2BlVUhsyf+i4oD7p/gpBPQTUPBFoVxMLpEwycub3YTHdqKpF9rCpEz76WLc3ODOWkCy3w
ysnoJ6iuNvH4XH9YpeukeQLrC3PdRhwf4+XrEQNGWosKNpk2Iy/QtUpPPt2gEUwws0owKd5XDKOtavG2qZBQ/+au
OK9H41A=="
```

[3] For release 3, the parameter to consider is "TPP-Certificate", instead of "TPP-Signature-Certificate" that will only be used on release 4.

4.4 Example of a signed message

The following data:

Private Key (in PEM format) that is the key pair of the public key included in the QSeal Certificate included in parameter TPP-Signature-Certificate	-----BEGIN PRIVATE KEY----- MIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQCfpjExqo7Ep33z Z9DzOUnPAVMX3hdpHIbQZh+fVpR9FI2HKjc96GzvuhEHLB1qjas9K0szPWsR1GCH nbFVFS4tLnbSz/TEA9tlIzmeAlJHRngbprhjAx0TLq6DCMikQoWacfde/m80p4ED Ha4moiQNr+mtekLyR4upiZ1N+fYCWJmYXNgtIlOQzhmol6gfhcfhReZtfgfJswDe 7K5Zqosa84BcfCa86SE41/Gnu2adzf4eUXDwj7YhkXZ6BsUgmM9hONqFBltnzjrU C2a1pCFI4LSc61gaRLZmXFMEoFV36f+Sc2jKO6e89OqKzhMoy8zOMjYh3CVO/vYT TUkfsncvAgMBAAECggEAC4oWqjFaymoU14AcgUgVxqmg2OvLo2RVdkC7bmfyqavP owJxJb9kCFvZmTweEDKXOv+jGtwInixMoeDLDYxPXyKpogk39ucUd5X5NyuRyOcW DWGamEWEV3ubT4tV/VF1mLk+GRao8RReZdxCVzaHBpo9eLWKmzqCpMl2nkk/9+FU VuzmoLpFmyPFK03bH/Nl5w92PYQ5ttkNOdqQzCxISOD/8BqU4KbV2Ng4gi82Xvvh VDL6HffYpujspi7DejbSBZ+bOsBe1ZQEhtzNNJ/Q4Uv8XIuLm2h2cm/E+bNfEr7z A8SkG7PpB+SRjSteaSE+SoeJ5RT7t6kgWXS1C/jL6QKBgQDLtUGaQEqKZTspvbTu CbA1Vp67LEoy5T4Mga2vob1KVvGmQEfuhiI1cj5/cgPGZ1yojYHDpnwzQbfFkPxi KFFgp4wnXGlRr0EeJYMZjW5G5Ms2M/Bcl3CvwaDL2sISjWt2FhxSQZQx39kM5m6N KvEN+ICBLqwMIon6em7b91DUFQKBgQDIoZf/F/lvtReYXUrl3spkviLNYkBbZozW YCz2riQHChXNupMsa4nFlb2+Y1goELUWOTELb0Jw1lybLewx9FFdFVncRAjgHGBA P5CsoJ2/9T/vi/N2Wj0J5DiPdfEGwpwOqXwBfh1lVlkgSnNTYXlIp+ANcDHyVhQH nf/1r9IbMwKBgBP/rXsZSLghjBdi+npMFTKHWHoDtR0eCGNt78FIXa8IrhymPumZ 3Y3ls2ELrncx+pTJn623kIXvs7z/qOdyEdstV8MdfXF5hSkSgbZmficTmyetHbHZ ZES8+65HwbnUjHJBZXJl4cirs0Fi9gOB7bxzYxpLnLRsR6OBlZSeyR6pAoGARIG5 Due2yogBeItSic9bOK8b3xmPdCY+LO1GLS0lLCora8Yrft1xe8A3vAzcC0I5MO9w CcGB5Fmt4Wb64cvVBH3H40hv52aJDyclVWy6sNMjc75L8bu6X+hHz+Sr2m0VMIR6 zV+s1e94G2iQnIYKDd8UyEHpLCBsUnWG8vOIQLUCgYAh7biRup4jhXmKc9TukCHI Z2h6PeOwFaFJcR4+3o+WTEuWy8+h1pPnlkEr8FvbLhl1B3B0hJJl0TzrP6p9FLHv Y1h+Ux1J4Il7LxVVhG5wZA3ahpxhw9vQBM+2xawoGG5Zi41kpBAWERRI8SlaqnxO ZcQpdIZkwnaXQT8uVuVjcg== -----END PRIVATE KEY-----

Message Body	{
	"access": {
	"balances": [{
	"iban": "PT50000000000000000000000",
	"currency": "EUR"
	}],
	"transactions": [{
	"iban": "PT50000000000000000000000",
	"currency": "EUR"
	}],
	"accounts": [{
	"iban": "PT50000000000000000000000",
	"currency": "EUR"
	}]
	},
	"recurringIndicator": true,
	"validUntil": "2019-12-01T13:20:00",
	"frequencyPerDay": 4,
	"combinedServiceIndicator": false
	}

Digest	SHA-256=LrIQs5UqJlzOX3B+wk25SEaUEalqvRMDCbrQFFKEaRs=

Signing String	digest: SHA-256= LrIQs5UqJlz0X3B+wk25SEaUEalqvRMDCbrQFFKEaRs= tpp-transaction-id: ceae5ddb2325457bac80b43baefaf558 tpp-request-id: bcd4aad6fcc246419485a015f4cb6996 psu-id: PSU-123 date: 2019-08-19T17:44:25.918+01:00

Results in the following POST API request message headers for release 3 (for results in release 4, the fields must be adapted according with the information provided above):

```
POST https://site1.sibsapimarket.com:8444/sibs/apimarket-sb/BST/v1-0-
2/consents?tppRedirectPreferred=false&withBalance=false HTTP/1.1
Accept-Encoding: gzip,deflate
PSU-ID-Type: sdfasdf
TPP-Redirect-URI: /teste/teste
Content-Type: application/json
x-ibm-client-id: 476bd8e5-e6f9-4f83-bb38-a07f56a063f2
TPP-Request-ID: bcd4aad6fcc246419485a015f4cb6996
Digest: SHA-256=LrIQs5UqJlz0X3B+wk25SEaUEalqvRMDCbrQFFKEaRs=
Signature: keyId="44ba0c31580926fcda18ddee8d1ac0a9",algorithm="rsa-
sha256",headers="Digest TPP-Transaction-ID TPP-Request-ID PSU-ID
Date",signature="QN5IfGeEvSWX0PYMuIsxfAsjFwEuLpxR4oKClhrxL77Fiuha9rDSrNPmKjk7eS
kIVQlSQrsHwMnLuzo9uvkA9fLTPYFRVSm6seiDWeGG8Gxo2SCIyDjvGQBHXyM2k3AQ7ChQy8IKq6Uvg
xASyUKCnOJpdp11y9aLvjwbzCiIr7dbQwXMd0Jb6yptOVWwm1g7OebN9js+zkFzotgjWtSMMwpkieA4
DSNL95JYmaOQxz4K8IEhgoN96pS66pnsp0ZI+1vOi70X+LPs3EP16AC+Qh7DGlc1iyagE8EQ0FhSYwW
gYEuaFfJFv518BAVVgankwKN6Kph9sZFta+vxYMZAxA=="
PSU-ID: PSU-123
TPP-Certificate:
MIIIRTCCBi2gAwIBAgIQRLoMMVgJJvzaGN3ujRrAqTANBgkqhkiG9w0BAQsFADCBuTELMAkGA1UEBhM
CUFQxQjBABgNVBAoMOU1VTFRJQ0VSVCAtIFNlcnZpw6dvcyBkZSBDZXJ0aWZpY2HDp8OjbyBFbGVjdH
LDs25pY2EgUy5BLjEgMB4GA1UECwwXQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkxRDBCBgNVBAMMOyhDR
VJUKSBNVUxUSUNFUlQgVHJ1c3QgU2VydmljZXMgQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkgMDA0MB4X
DTE5MDQwMzE0MzYwMFoXDTE5MDcwMzE0MzYwMFowgaoxCzAJBgNVBAYTAlBUMTswOQYDVQQKDDJTSUJ
TIC0gU29jaWVkYWRlIEludGVyYmFuY8OhcmlhIGRlIFNlcnZpw6dvcywgUy5BLjEWMBQGA1UEYQwNUF
NEUFQtQlAtOTk5OTE3MDUGA1UECwwuUFNEMiBRdWFsaWZpZWQgQ2VydGlmaWNhdGUgZm9yIEVsZWN0c
m9uaWMgU2VhbDENMAsGA1UEAwwEU0lCUzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAJ+m
MTGqjsSnffNnOPM5Sc8BUxfeF2kchtBmH59WlHOUjYcqNz3obO+6EQcsHWqNqz0rSzM9axHUYIedsVU
VLi0udtLP9MQD22UjOZ4CUkdGeBumuGMDHRMuroMIyKRChZpx917+bzSngQMdriaiJA2v6a16QvJHi6
mJnU359gJYmZhc2C0iU5DOGaiXqB+Fx+FF5m1+B8mzAN7srlmqixrzgFx8JrzpITjX8ae7Zp3N/h5Rc
PCPtiGRdnoGxSCYz2E42oUGW2fOOtQLZrWkIUjgtJzrWBpEtmZcUwSgVXfp/5JzaMo7p7z06orOEyjL
zM4yNiHcJU7+9hNNSR+ydy8CAwEAAaOCA1QwggNQMAwGA1UdEwEB/wQCMAAwHwYDVR0jBBgwFoAUpOp
A8d9T3pjwISxUIW3m7otTUtswgZwGCCsGAQUFBwEBBIGPMIGMMEoGCCsGAQUFBzAChj5odHRwczovL3
BraS50ZXN0ZS5tdWx0aWNlcnQuY29tL2NlcnQvTVVMVElDRVJUX0NBL1RTQ0FfMDA0LmNlcjA+BggrB
gEFBQcwAYYyaHR0cDovL29jc3AudGVzdGUubXVsdGljZXJ0LmNvbS9vY3NwLXNlcnZpY2VzL29jc3Aw
RwYDVR0uBEAwPjA8oDqgOIY2aHR0cDovL3BraS50ZXN0ZS5tdWx0aWNlcnQuY29tL2NybC9jcmxfdHM
wMDRfZGVsdGEuY3JsMGcGA1UdIARgMF4wCQYHBACL7EABATARBg8rBgEEAYHDbgEBAQEAAQ4wPgYNKw
YBBAGBw24BAQEABzAtMCsGCCsGAQUFBwIBFh9odHRwczovL3BraS50ZXN0ZS5tdWx0aWNlcnQuY29tM
IIBWgYIKwYBBQUHAQMEggFMMIIBSDAKBggrBgEFBQcLAjAIBgYEAI5GAQEwCwYGBACORgEDAgEHMBMG
BgQAjkYBBjAJBgcEAI5GAQYCMIGtBgYEAI5GAQUwgaIwTxZJaHR0cHM6Ly9wa2kudGVzdGUubXVsdGl
jZXJ0LmNvbS9wb2wvY3BzL01VTFRJQ0VSVF9QSi5DQTNfMjQuMV8wMDAxX2VuLnBkZhMCRU4wTxZJaH
R0cHM6Ly9wa2kudGVzdGUubXVsdGljZXJ0LmNvbS9wb2wvY3BzL01VTFRJQ0VSVF9QSi5DQTNfMjQuM
V8wMDAxX3B0LnBkZhMCUFQwXgYGBACBmCcCMFQwOTARBgcEAIGYJwEDDAZQU1BfQUkwEQYHBACBmCcB
BAwGUFNQX0lDMBEGBwQAgZgnAQIMBlBTUF9QSQwQQmFuayBvZiBQb3J0dWdhbAwFUFQtQlAwQQYDVR0
fBDowODA2oDSgMoYwaHR0cDovL3BraS50ZXN0ZS5tdWx0aWNlcnQuY29tL2NybC9jcmxfdHMwMDQuY3
JsMB0GA1UdDgQWBBQ1/ymvoiX0PE1JHfd9zeRBoApm1zAOBgNVHQ8BAf8EBAMCBkAwDQYJKoZIhvcNA
```

```
QELBQADggIBAI/X+DV/OzUHOCaFYjNBhbWhOY6uNoZEk/Slw9eCnuBAGLBKeiyCdFwWHeol5l5XZHvg
7re5yf78cSuYij6IJNJRKRJlg/1lgSKAlsFSRlsodXOYoO1kFJ33DsOmuGdX+ameD7/zsl2mT3pFn5j
nrv8TdZE1UtB3ce8mtWDwqXlLW9pQWhF9I6JgXUhqK9Bj9TxmlRBxAZIwAS22BFp6YjWTwValHqISg2
OXMVmRLVOIsyrAaeJm4PyfNY41jugREnOOGQJEpF6FMuFmckzZPAahRW+76GKdnl+3Vs7Xs6EtiozUr
u5NfsqKbbFH5WzRG8hFj4ZnKjlrO1y7xky9vXaTSlQKqEdd27KP9iGLZw/rFqXMO9WsKQlkU6gbjzJ8
wRg7tUygysz1oeZI5vEd5+iDBgbDqW5PGd+l3tKvPVBCo9pO8D+E6OBsXTtgNuyvooUm5eaNtKeujxW
n629DZZ2p1gNwbo1wkO76iQplju8vO3R3wwIakqHhorKLYCmtJ5y9xb49jTtAOaas/Zm1jckOPI9vxy
4QCczbEoYPLzBhJvSZbwJI2yrTtq+tP8lFuuJm2r/eO7WealGmjndo42orCDz31voxgmIIwx4P2/+Zc
L1XrSZl3xjpIxmmsrPiv9IYLHgHAznROwYMlOHpoYaDjgSB98EvfduVESSu+2gM
Date: 2019-08-19T17:44:25.918+01:00
TPP-Transaction-ID: ceae5ddb2325457bac80b43baefaf558
Content-Length: 575
Host: 172.23.133.52
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

5 Contingency procedures

SIBS API Market has been built to provide TPPs and ASPSPs with high availability and fault-tolerant operations. SIBS API Market infrastructure is divided into two redundant active/active sites.

TPPs may call API endpoints at either site at any time. In case one site becomes unresponsive, TPPs shall route all API calls to the other site to continue providing services.

TPPs should not notice any difference in the execution of API calls regardless of the site selected. All resources created in API calls (e.g., payment and consent resources) are shared between the two sites. Resource identifiers (e.g., "consentId" or "paymentId") created in one site can be used to address the same resource on the other site.

The choice of the SIBS API Market site that will process the required API operation is made via the {host} value in the API endpoint, as per section 2.2.

6 API flows

This section describes the order in which TPPs must perform each API operation to get the intended service.

The execution flow of each API depends on the authentication approach requested by the ASPSP to authenticate the PSU. The possible authentication approaches are:

Redirect	The steps for the PSU authentication are not performed at the interface between the SIBS API Market gateway and the TPP, but directly between the PSU and the ASPSP. In this case, the TPP shall redirect the PSU user agent (e.g.: web browser or app) to an ASPSP authentication web interface. The URL to this web interface is included in the ASPSP's response to the initial API call.
Decoupled	The steps for the PSU authentication are not performed at the interface between the SIBS API Market gateway and the TPP, but directly between the PSU and ASPSP. In this case, the ASPSP asks the PSU to authenticate on a separate channel, e.g., by sending a push notification with the transaction details to a dedicated mobile app or via any other application or device that the ASPSP makes available to the PSU.
Embedded	If the embedded approach is applied, PSU authentication is performed entirely as part of the transaction at the interface between the SIBS API Market gateway and the TPP. PSU authentication elements are gathered by the TPP and sent to the ASPSP for verification through the interface between the SIBS API Market gateway and the TPP.

Each ASPSP decides which authentication approach they will support and select on each transaction. Please refer to section 2.

The selection of the authentication approach is decided by the ASPSP during the initial POST API operation. In the response to the POST operation (and the following PUT operations, if requested), the ASPSP sends requests to the TPP, which depend on the selected authentication approach, using the "_links" parameter according to the API Steering Process by Hyperlinks defined in [BG-IG].

When the ASPSP stops sending requests in the "_links" parameter of POST and PUT operations responses, the TPP must issue the GET status API operation, to get information on the success of the API execution.

All PSD2 API requests must be addressed to one of the ASPSP available on SIBS API Market. The list of ASPSPs is available in section 2. The TPP selects the ASPSP recipient for each call to the API operation in the aspsp-cde path parameters of the operation endpoint. The aspsp-cde assigned to each ASPSP can be retrieved using the List of Banks API, available in the Information APIs product. The List of Banks API provides information on the ASPSPs that may be addressed on SIBS API Market. In addition to the aspsp-cde, this API provides information the TPPs may present to the PSU (e.g., logotype) during the selection of ASPSPs by the PSUs, as well as the list of APIs provided by each ASPSP. Based on the list of provided APIs, TPPs may build a dynamic list of logotypes to display to the PSU whenever the selection of ASPSPs by the PSU is needed, including in the list only the ASPSPs that provide the required service (for example, during the execution of a payment initiation using the instant-sepa-credit-transfers payment product, TPPs may filter the list of ASPSPs displayed to the PSU for selection of their ASPSP, so that it includes only the ASPSPs that show the payments/instant-sepa-credit-transfers API in the list of APIs provided by the List of Banks API).

For simplicity, the protocol (https://) and the {path} are omitted in the API endpoints used in the flows below.

6.1 Payment Initiation

This API initiates a transfer of funds from a PSU's payment account, held by the PSU in one of the ASPSPs available in SIBS API Market, to a beneficiary's account. The list of ASPSPs is available in section 2. A Payment Initiation request addressed to an ASPSP with an IBAN pattern that does not belong to that ASPSP will be rejected immediately (e.g., 'DE89' for an ASPSP based in Portugal). Depending on the payment product, the day of the week and the time, the payment may be settled immediately, and the funds credited to the beneficiary's account, or may be scheduled for settlement during the next settlement routine of the ASPSP. The transaction status returned by the ASPSP provides information on the payment execution.

The payment product used for the transfer of funds (e.g., SEPA Credit Transfers, SWIFT for international credit transfers) is defined by the payment-product parameter included in the API endpoint path. The list of payment products available per ASPSP can be found in section 2 and, preferably, through the List of Banks API.

The Payment Initiation flow ends when the ASPSP returns a final transaction status in the GET payment status API response. The transaction status is sent in "transactionStatus" parameter.

Payment resources are deleted, and the response '404-Not found' is returned:

- one month after the payment initiation has reached a final status;
- 30 minutes after the ASPSP fails to perform PSU authentication in the redirect and decoupled approaches (e.g., the PSU browser redirection or the push notification to the dedicated app did not work, or the PSU abandoned the authentication process).

While the final status of the transaction is not returned in the GET payment status API response, the TPP may issue the GET payments status API operation until a final status is returned. In the redirect and embedded authentication approaches, the call to the GET payment status API is performed after PSU authentication is completed, and a final status may be returned immediately, whereas in the decoupled authentication approach the TPP needs to poll the ASPSP to find out when the PSU authentication has ended. To prevent excessive bandwidth consumption, which could jeopardize the stability of the service, SIBS API Market implements throttling mechanisms. It is recommended to allow at least a 5-second delay between calls to the GET payment status API during status polling.

Once a payment initiation reaches a final status, no more changes will be made to the payment initiation status until the payment resource is deleted (e.g., if the returned status is "ACSC", all following GET status will return "ACSC", even if the payment is sent to settlement by the ASPSP during the 30 minutes before the resource is deleted).

The possible values for "transactionStatus"/code parameter, and the definition of the final status, are included in the following table:

Table 4 – Payment Initiation - Possible values for "transactionStatus"/code parameter

"transactionStatus"	Status	Definition
RCVD	Received	The ASPSP has received the payment initiation request. This is an initial status.
PDNG	Pending	The ASPSP is performing PSU authentication. Further verifications and status updates will be performed. This is an intermediate status.
PATC	Partially Accepted Technical Correct	The ASPSP has successfully authenticated the PSU. The payment initiation has been accepted, but the funds transfer policy for the account requires the authorisation of other accountholders (typically on corporate accounts). The remaining authorisations will be gathered by the ASPSP on his client direct interfaces (e.g., home banking). Once all the required authorisations are granted, the payment initiation status evolves to a final status. This is an intermediate status.
RJCT	Rejected	The PSU has refused the payment or failed the authentication, or an error has occurred. This is a final status.
CANC	Cancelled	The payment has been cancelled by the TPP or the PSU through the TPP. This is a final status.
ACTC	Accepted Technical Validation	The ASPSP has successfully authenticated the PSU. The availability of funds on the account has not been checked yet. The payment is scheduled to be booked in the PSU's account and sent for settlement at the ASPSP's next settlement routine. This is a final status for the recurring/periodic payments and future-dated payments.
ACSP	Accepted Settlement in Process	The payment has been booked in the PSU's account. All preceding checks, such as technical validation and customer profile, were successful, and therefore the payment initiation execution has been accepted. This is a final status.
ACSC	Accepted Settlement Completed	The payment has been booked and settled in the PSU's account. Depending on the creditor agent, funds may have been credited to the beneficiary's (creditor's) account. This status will be seldom seen and may only occur if the ASPSP not only sends the payment for settlement but also receives a response from the settlement system after the successful PSU authentication and before the reception of the Get payment status API call from the TPP. This is a final status.
ACCC	Accepted Settlement Completed	Settlement in the creditor's account has been completed. This is a final status.

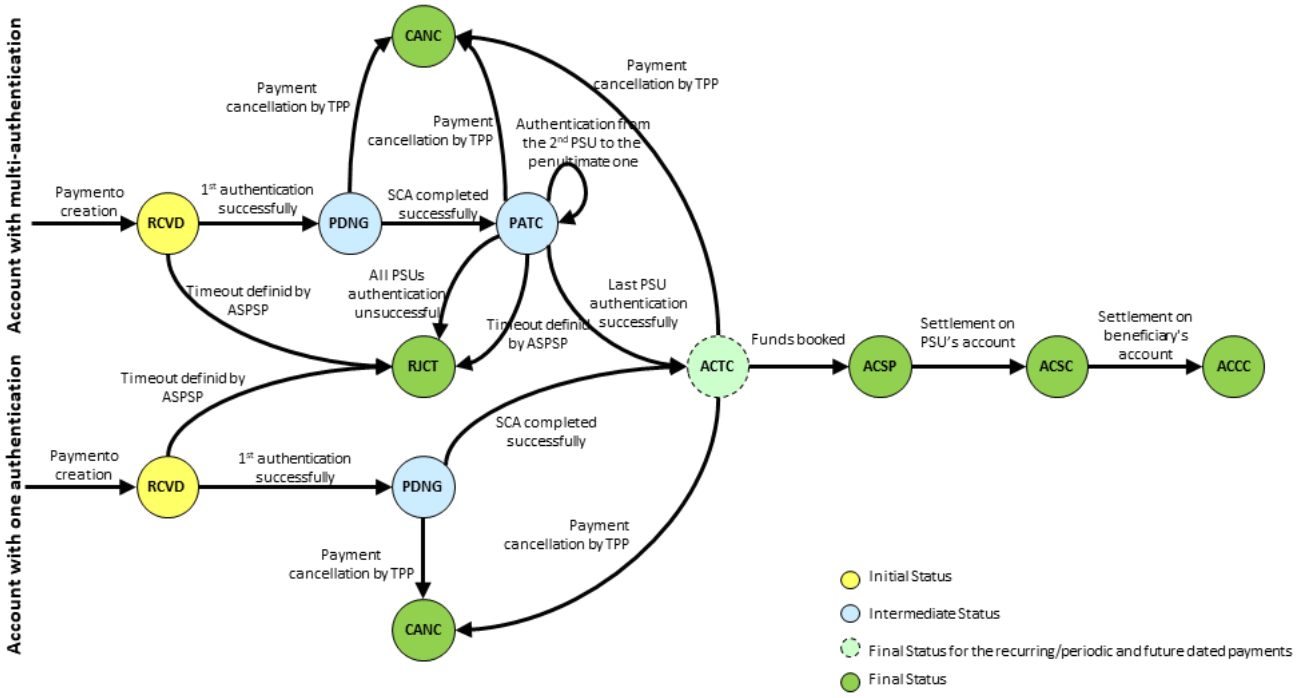


Figure 1 – Payment Initiation status diagram (with one authentication or multi-authentication)

6.1.1 Redirect flow

This is the 'happy path' flow for performing a payment initiation using the redirect authentication approach.

The ASPSP informs the TPP that a redirect flow must be performed by sending, in the POST operation response, the "_links" parameter containing the sub-parameter "redirect" with a URL to the ASPSP authentication web/app interface, to where the TPP shall redirect the PSU's user agent: "links": {"redirect": "URL_of_the_authentication_web_interface"} – Implicit creation of authorisation resource.

This is the default mechanism for most ASPSPs. Please note that, in particular for authorisation creation, the vast majority of ASPSPs only require authorisation by one PSU and therefore prioritise the implicit creation of the authorisation resource.

Once the ASPSP has finished PSU authentication, they redirect the PSU's user agent back to the TPP's payment completion interface. The URL of the TPP's web interface is provided to the ASPSP in the "TPP-Redirect-URI" parameter in the initial POST payment operation. In the path or query parameters of this URL, the TPP must include elements that allow their payment completion web interface to identify the transaction upon redirection of the PSU's user agent. The URL must not include any sensitive information. The transaction identification elements should be non-reusable, randomly generated, and large enough to make guessing a valid value virtually impossible. If the PSU does not enter any access credentials but instead selects "cancel," they must be redirected using the link provided by the TPP in the "TPP-Nok-Redirect-URl" parameter.

If an authorisation resource is created explicitly, no authorisation ID is required, so the ASPSP will send the 'start authorisation' link to the TPP in order to proceed. The authorisation link will only appear in the second phase of the flow.

Some ASPSPs may not offer the explicit creation of the authorisation resource, especially at an early stage of release 4 implementation.

Once the PSU's user agent reaches the TPP's payment completion web interface, the TPP may issue the GET payment status API operation to retrieve information on the PSU authentication result and the payment initiation request completion, via the "transactionStatus" parameter.

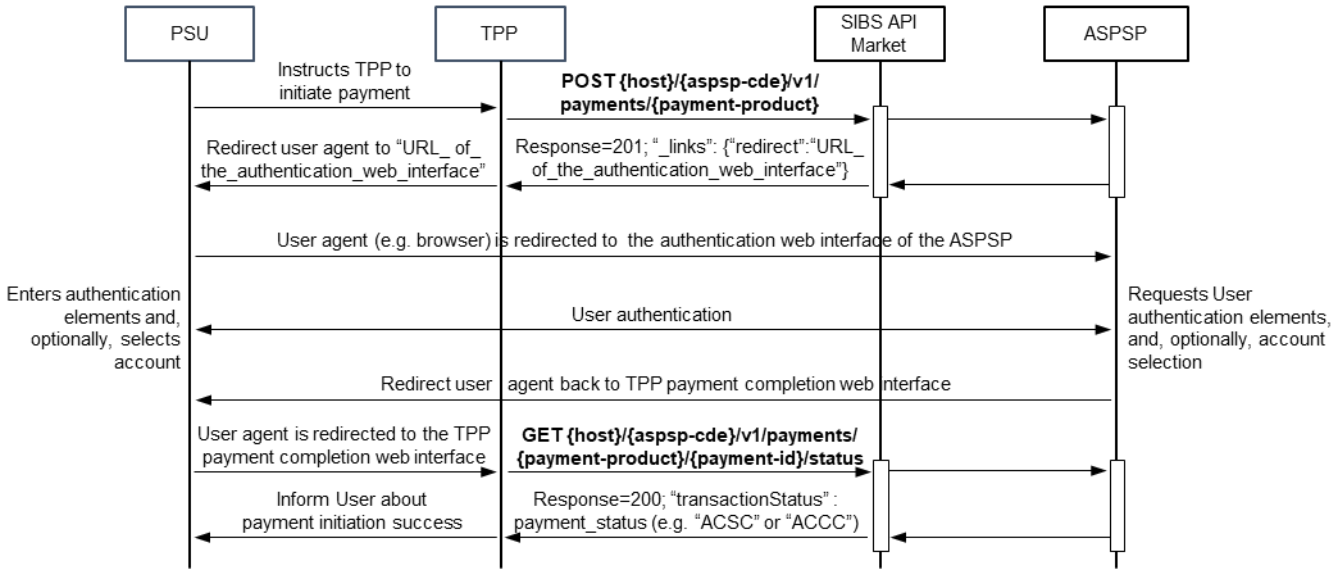


Figure 2 - Payment Initiation flow for the redirect authentication approach

6.1.2 Decoupled flow

This is the 'happy path' flow for performing a payment initiation using the decoupled authentication approach.

In the decoupled flow, the ASPSP needs to receive the User Identification used by the PSU to identify themselves on the ASPSP channels (e.g., home banking). For ASPSPs that have only implemented the decoupled flow, TPPs can request the User Identification from the PSU and send it in the initial POST payment initiation request. If the User Identification is not provided by the TPP in the POST operation, the ASPSP requests it by sending, in the POST response, the "_links" parameter containing the sub-parameter "updatePsuIdentification" with a URL to the endpoint that the TPP should use in the PUT operation to update the User identification. The TPP then requests that the PSU enters the identifier and sends it to the ASPSP in the "PSU-ID" parameter of the PUT operation.

Whenever the parameter "psuMessage" is sent by the ASPSP in the response to the POST and PUT API operations, the TPP should present its content to the PSU, as it may provide information and guidance to them (e.g., informing the PSU of the User Identification they need to enter for the

"updatePsuIdentification" request, or providing guidance to the PSU on using the dedicated app for authentication).

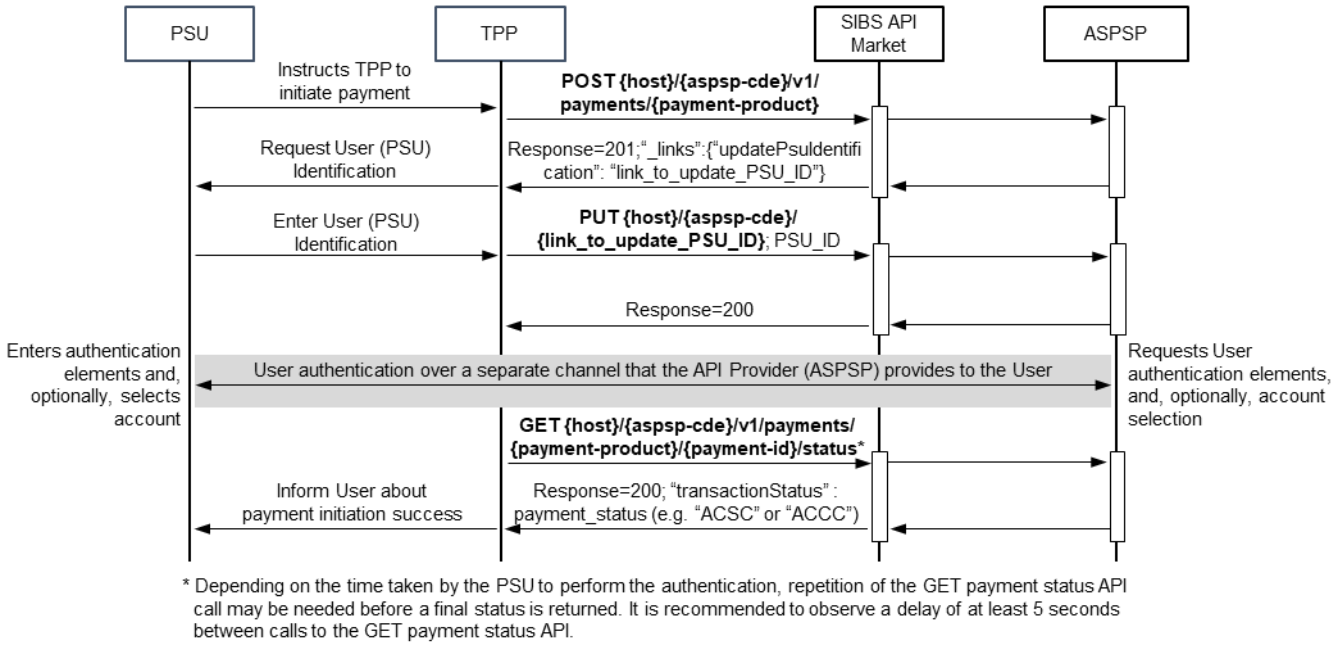


Figure 3 - Payment Initiation flow for the decoupled authentication approach

6.1.3 Embedded flow

This is the 'happy path' flow for performing a payment initiation using the embedded authentication approach.

In the embedded flow, all the authentication elements needed by the ASPSP to authenticate the PSU are exchanged through the APIs in a chain of successive update requests sent by the ASPSP to the TPP, until the API Provider (ASPSP) receives all the elements. The ASPSP requests the authentication elements by sending, in the POST and PUT responses, the "_links" parameter containing sub-parameters that inform the TPP which element is required and the URL of the endpoint that the TPP should use with the PUT operation to update the authentication data in the ASPSP (e.g., "_links":{"updatePsuAuthentication":"aspsp-cde/v1/payments/payment-product/ payment-id"). In addition to the "_links" parameter, the ASPSP may send further parameters to provide the TPP with information on the data to be requested from the PSU. Please refer to [BG-IG] for the complete list of values for the "_links" parameter and additional parameters (e.g., "chosenScaMethod" and "challengeData").

The authentication data required to authenticate the PSU depends on the ASPSP.

Whenever the parameter "psuMessage" is sent by the ASPSP in the response to POST and PUT API operations, the TPP should display its contents to the PSU, as these may provide the PSU with information and guidance (e.g., informing the PSU of the User Identification they need to enter for the "updatePsuIdentification" request).

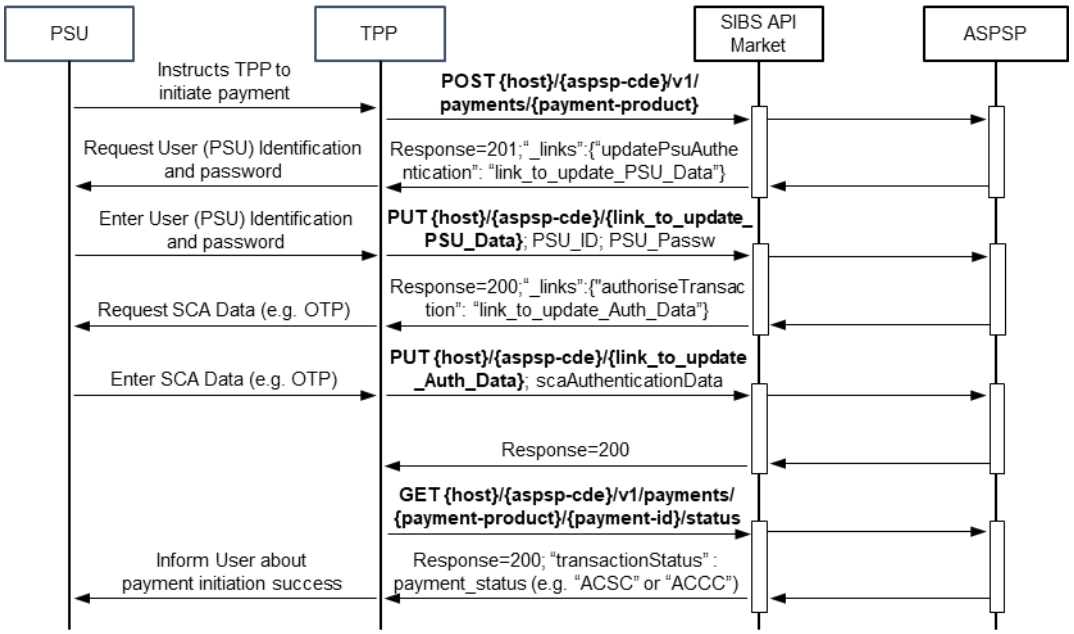


Figure 4 - Payment Initiation flow for the embedded authentication approach

6.2 Consent, account and card account information

This APIs allow a TPP to access account or card identification details (e.g., "IBAN" or "PAN"), balances and statements of one or more accounts held by the PSU in one of the ASPSPs available on SIBS API Market. The list of available ASPSPs can be found on section 2.

The flow shown in this section shall be repeated for each ASPSP where the PSU holds a payment account to be accessed by the TPP.

The Consent, Account and Card Account Information APIs must be used together to receive account/card identification details (e.g.: "IBAN" or "PAN"), balances and statements.

The Consent API allows a PSU to grant the TPP access to, or the ASPSP permission to provide, information of their payment accounts/cards. Once the Consent API has been successfully completed, the TPP will be given a Consent Identification, which is used in the Account Information/Card Account API to obtain account identification details (e.g.: "IBAN" or "PAN"), balances and statements.

When requesting these statements, the TPP should note that each transaction may have a Transaction ID. The management of the Transaction ID in the list of transactions is at the discretion of each ASPSP. Some ASPSPs do not have a unique transaction ID available for each transaction in their systems.

The Transaction ID in the first transaction of the GET transactions response is required for pagination and is unique (a sequence of concatenated transaction messages). Therefore, TPPs should expect ASPSPs to send a Transaction ID for each transaction.

The TPP may request one-off ('False') or long-lasting ('True') consent via the "recurringIndicator" parameter. When the TPP sets this parameter to 'False', the PSU must always be involved in the operation. One-off consents are valid for 30 minutes following their creation. During this period, there is no limitation on the number of API calls to retrieve information within the scope of the consent, and the transaction history is similar to that available to users when accessing the ASPSP channel directly and may vary among ASPSPs. Long-lasting consents are valid for a maximum of 180 days[4], and the validity period depends on the API provider (see next paragraph).

To comply with Article 10 of the [RTS] for the SCA exemption on account information, long-lasting consents are created with an expiry date. One month after this date, the consent resources are deleted. Any attempt to access a deleted consent resource will result in an HTTP response code of 404 – Not Found. While the TPP may include a 'Valid Until' date in the POST consent API request, the ASPSP may anticipate the expiration date for the consent according to their SCA exemption on account information policy, which is common to all the channels provided by the ASPSP to the PSU (e.g., home banking).

The Consent API provides the parameter "access" for defining the consent scope. Depending on this parameter and the PSU's decision, the consent scope may include a list of accounts, cards, balances and transactions.

The following use cases provide examples of the usage of the "access" parameter:

Table 5 – Use cases on the usage of the "access" parameter

ID	Use Case Description	Content of "access" parameter
#1	TPP needs to get the list of all payment accounts and cards held by the PSU at the ASPSP. The PSU gives consent to the TPP.	`{"availableAccounts":"all-accounts"}`
#2	TPP needs to get the list of all payment accounts and cards held by the PSU at the ASPSP, as well as the Account Holder Names of those accounts/cards. The PSU gives consent to the TPP.	{"availableAccounts":"all-accounts-with-ownerName"}
#3	TPP needs to get the list of all payment accounts and cards held by the PSU at the ASPSP, as well as the balances and transactions of those accounts/cards. The PSU gives consent to the TPP.	`{"allPsd2":"all-accounts"}`
#4	TPP needs to get the list of all payment accounts and cards held by the PSU at the ASPSP, as well as the balances, transactions and Account Holder Names of those accounts/cards. The PSU gives consent to the TPP.	{"allPsd2":"all-accounts-with-ownerName"}
#5	TPP needs to get the list of all payment accounts held by the PSU at the ASPSP, but the PSU does not grant consent to the TPP. In this case, the TPP suggests that the PSU selects the accounts on the ASPSP side through the same interface provided by the API Provider for PSU authentication.	`{"accounts":[]"}`

#6	TPP needs to get the list of all payment accounts held by the PSU at the ASPSP, but the PSU does not grant consent to the TPP.
	In this case, the TPP suggests that the PSU enters the IBANs of the accounts to which they grant the TPP access.
	The PSU enters IBAN_1 and IBAN_2.
		`{"accounts": [`
		`{ "iban": "IBAN_1" },`
		`{ "iban": " IBAN_2" }`
		`] }`

[4] This period changed from 90 days to 180 days upon entry into force of the revised article 10 of RTS on SCA.

ID	Use Case Description	Content of "access" parameter

#7	TPP needs to get the list of all payment accounts held by the PSU at the ASPSP, as well as the balances, transactions and Account Holder Names of those accounts, but the PSU does not grant consent to the TPP.
	In this case, the TPP suggests that the PSU grants the TPP consent to access balances, transactions and Account Holder Names, and to select the accounts on the ASPSP side through the same interface provided by the API Provider for PSU authentication
		{
		"balances": [],
		"transactions": []
		"additionalInformation": []
		}

#8	TPP needs to get the list of all payment accounts held by the PSU at the ASPSP, as well as the balances, transactions and Account Holder Names of those accounts, but the PSU does not grant consent to the TPP. In this case, the TPP allows the PSU to enter the account IBANs and, for each IBAN, to select the type of information to which they grant the TPP access: balances and/or transactions and/or Account Holder Names. The PSU enters: • IBAN_1 and only grants access to balances for this IBAN; • IBAN_2 and only grants access to transactions for this IBAN; • IBAN_3 and grants access to balances and transactions for this IBAN; • IBAN_4 and grants access to balances, transactions and Account Holder Names for this IBAN.	{ "balances": [{ "iban": "IBAN_1" }, { "iban": "IBAN_3" }, { "iban": "IBAN_4" }], "transactions": [{ "iban": "IBAN_2" }, { "iban": "IBAN_3" }, { "iban": "IBAN_4" }], "additionalInformation": { "iban": "IBAN_4" }] }
#9	TPP needs to get the list of all payment card accounts held by the PSU at the ASPSP, but the PSU does not grant consent to the TPP. In this case, the TPP suggests that the PSU selects the cards on the ASPSP side through the same interface provided by the API Provider for PSU authentication.	{"cards-accounts":[]"}
#10	TPP needs to get the list of all payment card accounts held by the PSU at the ASPSP, but the PSU does not grant consent to the TPP. In this case, the TPP suggests that the PSU enters the PANs of the cards to which they grant the TPP access. The PSU enters PAN_1 and PAN_2.	{"cards-accounts": [{ "pan": "PAN_1" }, { "pan": "PAN_2" }] }

#11	TPP needs to get the list of all payment card accounts held by the PSU at the ASPSP, as well as the balances, transactions and Account Holder Names of those accounts, but the PSU does not grant consent to the TPP.
	In this case, the TPP suggests that the PSU grants them access to balances, transactions and Account Holder Names, and selects the cards on the ASPSP side through the same interface provided by the API Provider for PSU authentication.
		{
		"balances": [],
		"transactions": []
		"additionalInformation": []
		}

#12	TPP needs to get the list of all payment card accounts held by the PSU at the ASPSP, as well as the balances, transactions and Account Holder Names of those cards, but the PSU does not grant consent to the TPP. In this case, the TPP allows the PSU to enter the card PANs and, for each PAN, to select the type of information to which they consent the TPP access: balances and/or transactions and/or Account Holder Names. The PSU enters: • PAN_1 and only grants access to balances for this IBAN; • PAN_2 and only grants access to transactions for this PAN; • PAN_3 and grants access to balances and transactions for this PAN; • PAN_4 and grants access to balances, transactions and Account Holder Names for this IBAN.	{ "balances": [{ "pan": "PAN_1" }, { "pan": "PAN_3" }, { "pan": "PAN_4" }], "transactions": [{ "pan": "PAN_2" }, { "pan": "PAN_3" }, { "pan": "PAN_4" }], "additionalInformation": { "pan": "PAN_4" }] }

ID	Use Case Description	Content of "access" parameter
#13	TPP needs to get the list of all payment accounts and card accounts held by the PSU at the ASPSP, as well as the balances and transactions of those accounts and cards, but the PSU does not grant consent to the TPP. In this case, the TPP allows the PSU to enter the account and card IBAN and PANs and, for each IBAN and PAN, to select the type of information to which they grant the TPP access: balances and/or transactions. The PSU enters: • IBAN_1 and only grants access to balances for this IBAN. • IBAN_3 and only grants access to transactions for this IBAN. • PAN_3 and grants access to balances and transactions for this PAN.	{ "balances": [{ "iban": "IBAN_1" }, { "pan": "PAN_3" }], "transactions": [{ "pan": "PAN_3" }, { "iban": "IBAN_3" }] }

Once a consent has been created, it is not possible to change the accounts/cards covered by the consent, nor the associated information (e.g., account name). If the TPP wants to change the list of accounts or card accounts (e.g., to add a new account/card), a new consent must be created containing the updated list of accounts/card accounts. The TPP may issue a POST consent request using the IBANs/PANs they received for the previous consent with the required changes or simply start again.

Successful creation of a long-lasting consent revokes any previous long-lasting consents that may exist for the same PSU, TPP and ASPSP. The unique key used to verify the uniqueness of the consent comprises the following parameters:

- aspsp-cde;
- TPP Registration Number (retrieved from the TPP eIDAS Certificate);
- PSU-ID;
- PSU-ID-Type.

Or, for corporate accounts:

- aspsp-cde;
- TPP Registration Number (retrieved from the TPP eIDAS Certificate) ;
- PSU-Corporate-ID;
- PSU-Corporate-ID-Type.

The TPP may agree with the PSU on the number of daily accesses, without PSU involvement, for retrieving account information under the consent scope, and send this number in the POST consent request in parameter "frequencyPerDay". According to [RTS], the ASPSP limits this number to 4. Once the number of accesses exceeds the value sent in the parameter "frequencyPerDay" (or 4 if "frequencyPerDay" is above 4), access to account information will be denied (HTTP status code 429 – Too Many Requests). This limit is defined per account under the consent scope, and per data set (balances and transactions). This limit does not apply if the PSU is participating in real time in the information request (this is indicated in the parameter "psuInvolved").

The consent flow ends when the ASPSP returns the final transaction status in the GET consent status API response. The transaction status is sent in the parameter "transactionStatus".

Consent resources are deleted, and a '404-Not found' response is returned:

- one month after the expiry date of the consent;
- 30 minutes after the response to the POST consent request, if the ASPSP has failed to perform PSU authentication in the redirect and decoupled approaches (e.g., PSU browser redirection or the push notification to the dedicated app did not work, or the PSU abandoned the authentication process).

A deprecated consent (RJCT) may remain in the database for a month before being deleted.

The TPP can perform the GET consent status API operation until a final status is returned. While in the redirect and embedded authentication approaches the call to the GET consent status API is performed after PSU authentication has ended, and a final status may be returned immediately, in the decoupled authentication approach the TPP needs to poll the ASPSP to find out when the PSU authentication has ended. To prevent excessive bandwidth consumption, which could jeopardise the stability of the service, SIBS API Market implements throttling mechanisms. It is recommended that at least a 5-second delay is observed between calls to the GET consent status API during the status polling.

Once a consent reaches a final status, no further changes will be made to its status until the consent resource expires/is revoked/deleted (EXPD/RVKD/TERM).

The possible values for the "transactionStatus"/code parameter and the definition of the final status are included in the following table:

Table 6 - Consent, account and card account information - Possible values for the "transactionStatus"/code parameter

"transactionStatus"	Status	Definition
RCVD	Received	The ASPSP has received the consent request. This is an initial status.
RJCT	Rejected	The PSU has refused the consent or failed the authentication, or an error has occurred. This is a final status.
PATC	PartiallyAuthorised	The ASPSP has successfully authenticated the PSU. The consent has been accepted, but the account information access policy for that account requires the authorisation of other accountholders (typically on corporate accounts). The remaining authorisations will be gathered by the ASPSP on their client direct interfaces (e.g., home banking). Once all the required authorisations are granted, the consent status evolves to a final status. This is an intermediate status.
VALD	Valid	The ASPSP has successfully authenticated the PSU. This is a final status.

"transactionStatus"	Status	Definition
RVKD	RevokedByPSU	Consent was revoked at the request of the PSU, which was duly reported through the ASPSP interface. This is a final status.
EXPD	Expired	Consent has expired. This is a final status.
TERM	TerminatedByTPP	The TPP terminated/cancelled the consent by selecting the DELETE option. This is a final status.

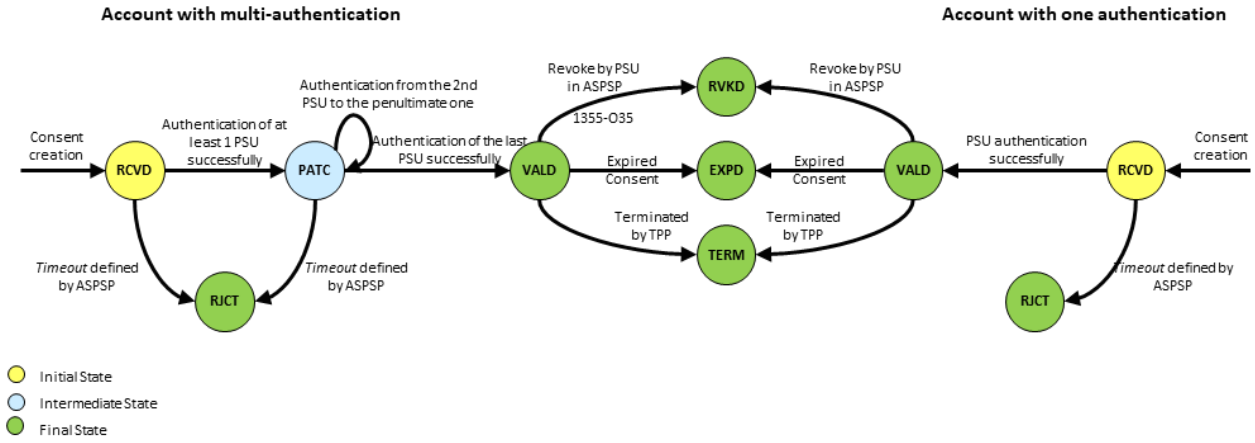


Figure 5 - Consent status diagram (with one authentication or multi-authentication)

Once the GET consent status API operation returns the "VALD" transaction status, the TPP may start requesting account information.

Account information requests will be denied if the TPP has not performed the GET consent status and received the "VALD" transaction status.

Before retrieving balances and transactions for each account/card under the consent scope, the TPP must obtain the links to use on the GET account details, balances and transactions operations for each account.

The GET accounts/card-accounts API operation returns the list of accounts covered by the consent and, for each account, the following parameters:

"id"	Account identifier. This account identifier is purely technical and is generated at random. May be used in the "account-id" path parameter of GET accounts/{account-id}, accounts/{account-id}/balances and accounts/{account-id}/transactions
"viewBalances" (in "_links" parameter)	URL to use with the GET operation to read account balances.
"viewTransactions" (in "_links" parameter)	URL to use with the GET operation to read account transactions.

The GET accounts/{account-id} operation, which is used to read account details, is denied unless the consent resource has been created with "allPsd2":"all-accounts", "accounts": [“iban”: "IBAN"] or "card-accounts": [“pan”: "PAN"] in the “access” parameter.

The GET accounts API operation returns all accounts, and the GET card-accounts operation returns all cards, that are covered by the consent, as long as the account/card account has been selected for at least one access type: details, balances or transactions.

If the response to the GET transactions operation is unable to include all the transactions booked and/or pending between the “dateFrom” and the “dateTo” parameters, SIBS API Market shall provide an URL for the TPP to retrieve more transactions (sub-parameter “next” within parameter “_links”). The TPP must perform the GET operation on the received URL without changing the URL (e.g., must not add query parameters), until the “next” sub-parameter is no longer included in the response.

According to [RTS], ASPSPs cannot exempt access to transactions older than 90 days from SCA. The GET transactions API operation is denied if the starting date, included in parameter “dateFrom”, is more than 90 days before the current date, unless consent has been created and SCA performed within the last 30 minutes. If the consent is more than 30 minutes old, then TPPs must create a new consent and issue the GET transactions operation within the next 30 minutes to retrieve transactions older than 90 days. The scope (i.e. the list of accounts and information to be retrieved) of the previous consent may be reused for the new consent to avoid requesting the PSU to define the consent scope again.

6.2.1 Redirect flow

This is the ‘happy path’ flow for creating a consent using the redirect authentication approach.

The ASPSP informs the TPP that a redirect flow shall be performed by sending in the POST operation response the “_links” parameter containing the sub-parameter “redirect” with a URL to the ASPSP authentication web/app interface, to where the TPP shall redirect the PSU’s user agent (e.g.: “_links”: {“redirect”: “URL_of_the _authentication_web_interface”}) – Implicit creation of authorisation resource.

This is the default mechanism for most ASPSPs. Please note that, in particular for authorisation creation, vast majority of ASPSPs only require authorisation by one PSU and therefore prioritise the implicit creation of the authorisation resource.

Once the ASPSP has finished PSU authentication, they redirect the PSU’s user agent back to the TPP’s consent completion web interface. The URL of the TPP’s web interface is provided to the ASPSP in the “TPP-Redirect-URI” parameter in the POST consents operation. In the path or query parameters of this URL, the TPP must include elements that allow their consent completion web interface to identify the transaction upon redirection of the PSU’s user agent by the ASPSP. The URL must not include any sensitive information. The transaction identification elements should be non-reusable, randomly generated, and large enough to make guessing a valid value virtually impossible. Also, these elements should not be valid for more than the necessary time (e.g.: 30 minutes).

If an authorisation resource is created explicitly, no authorisation ID is required, so the ASPSP will send the 'start authorisation' link to the TPP in order to proceed. The authorisation link will only appear in the second phase of the flow.

Some ASPSPs may not offer explicit creation of the authorisation resource, especially at an early stage of release 4 implementation.

Once the PSU's user agent reaches the TPP's consent completion web interface, the TPP may issue the GET consent status API operation to retrieve information on the PSU authentication result and the consent request completion, via the "transactionStatus" parameter.

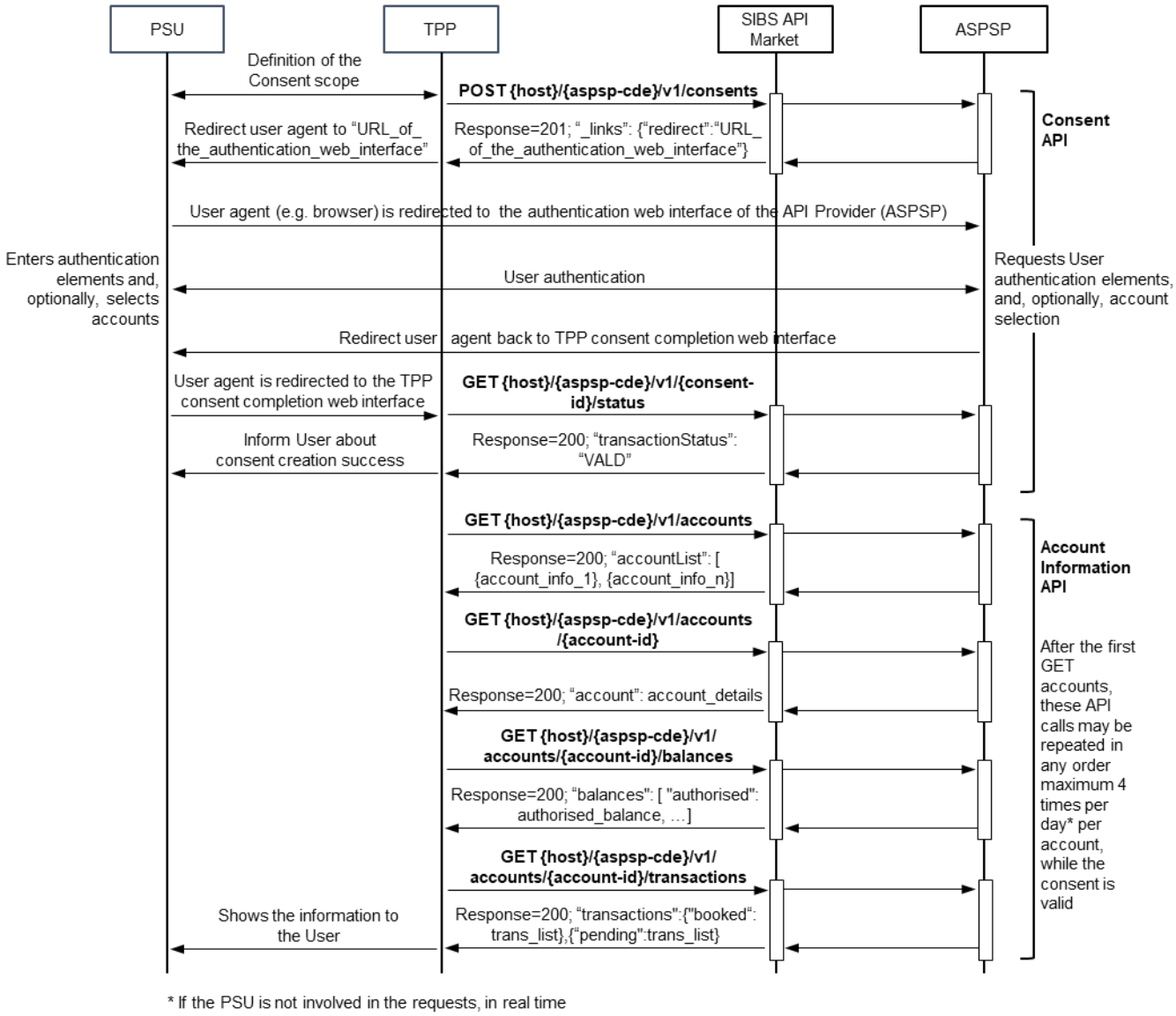


Figure 6 - Consent creation and account information flow for the redirect authentication approach

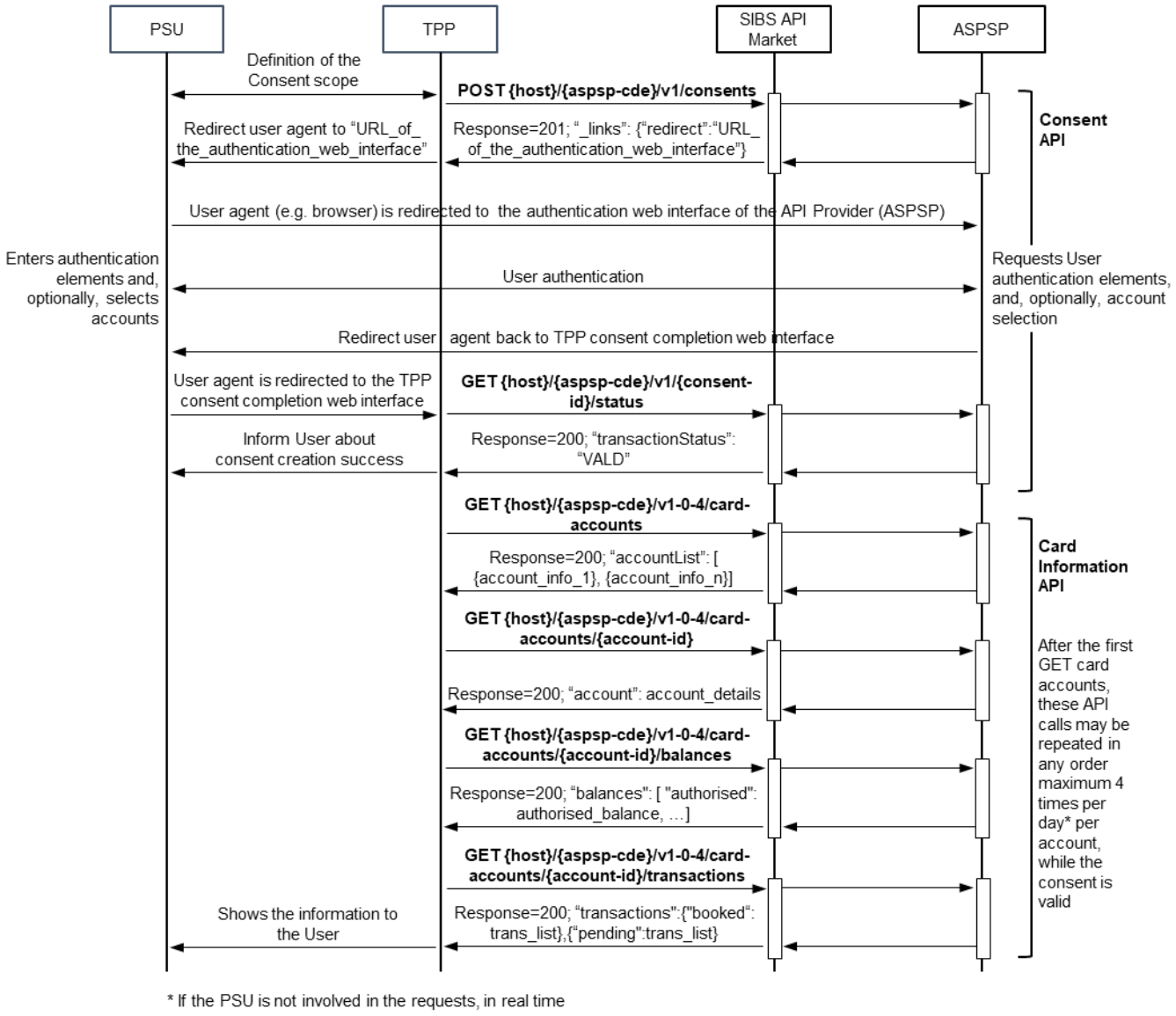


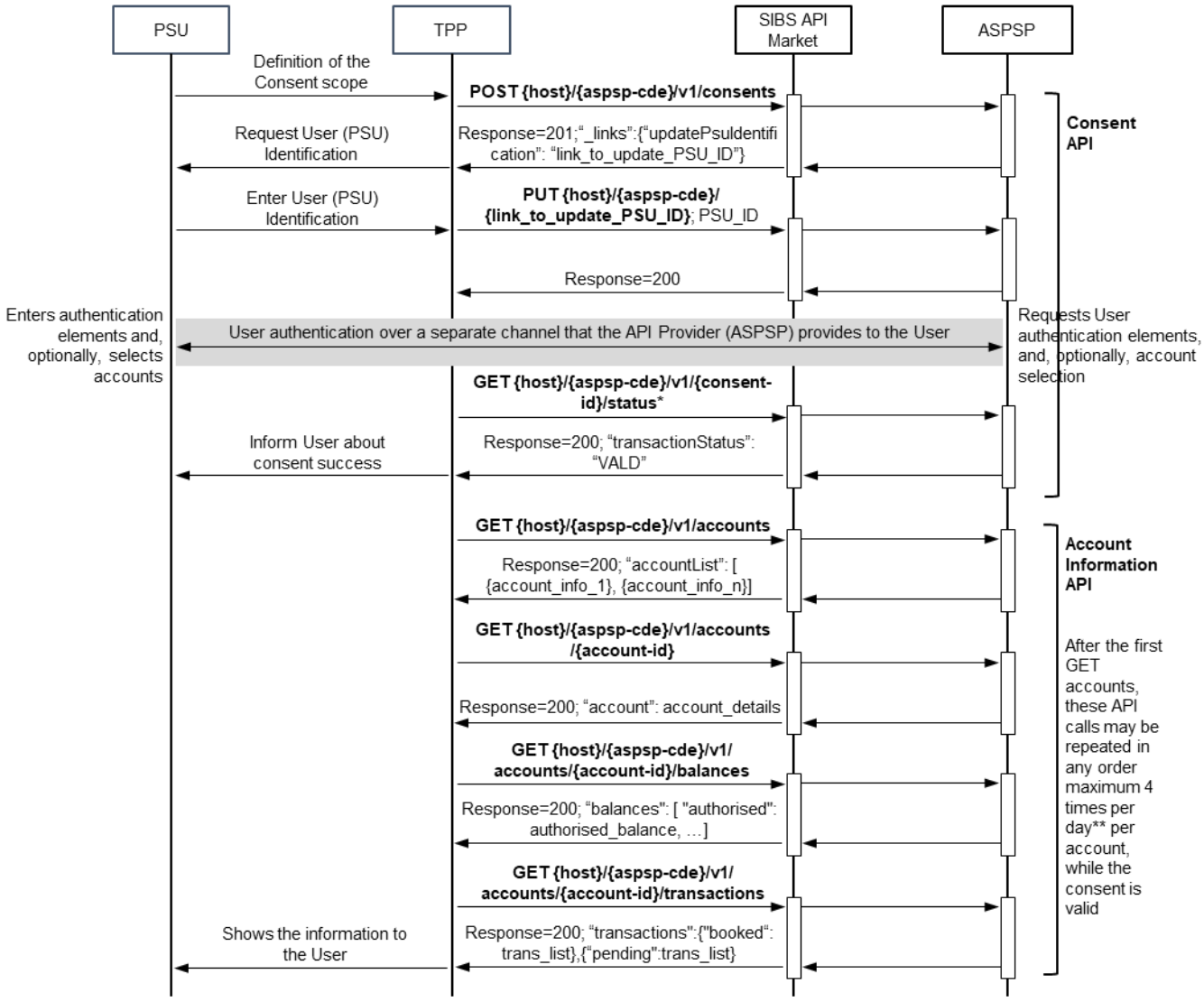
Figure 7 - Consent creation and card-account information flow for the redirect authentication approach

6.2.2 Decoupled flow

This is the 'happy path' flow for creating a consent using the decoupled authentication approach.

In the decoupled flow, the ASPSP needs to receive the User Identification used by the PSU to identify themselves on the ASPSP channels (e.g., home banking). For ASPSPs that have only implemented the decoupled flow, TPPs can request the User Identification from the PSU and send it in the initial POST consent request. If the User Identification is not provided by the TPP in the POST operation, the ASPSP requests it by sending, in the POST response, the "_links" parameter containing the sub-parameter "updatePsuIdentification" with a URL to the endpoint the TPP should use in the PUT operation to update the User identification. The TPP then requests that the PSU enters the identifier and sends it to the ASPSP in the "PSU-ID" parameter of the PUT operation.

Whenever the parameter "psuMessage" is sent by the ASPSP in the response to the POST and PUT API operations, the TPP should present its content to the PSU, as it may provide information and guidance to them (e.g., informing the PSU of the User Identification they need to enter for the "updatePsuIdentification" request, or providing guidance to the PSU on using the dedicated app for authentication).



* Depending on the time taken by the PSU to perform the authentication, repetition of the GET payment status API call may be needed before a final status is returned. It is recommended to observe a delay of at least 5 seconds between calls to the GET payment status API.
** If the PSU is not involved in the requests, in real time

Figure 8 - Consent creation and account information flow for the decoupled authentication approach

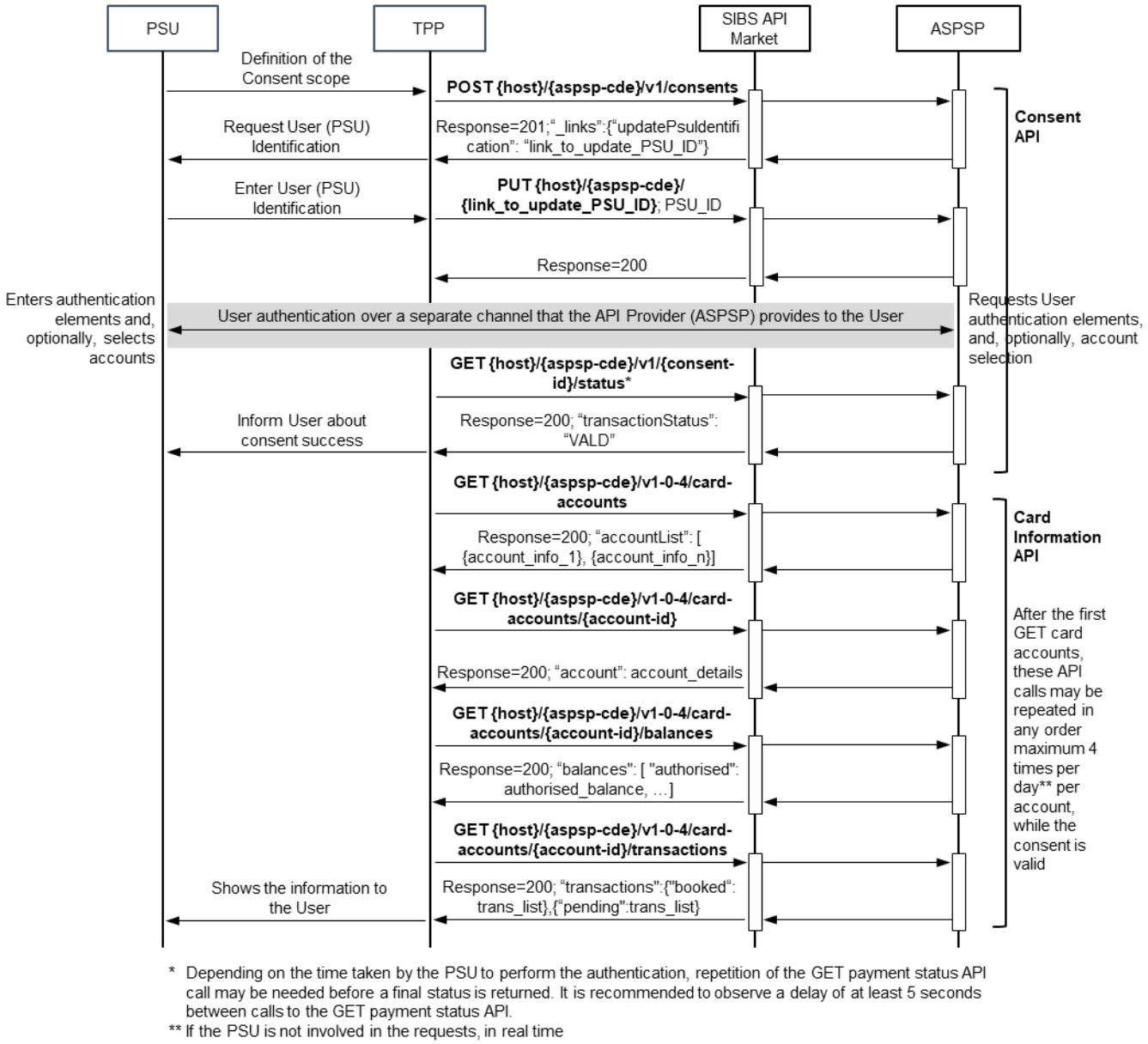


Figure 9 - Consent creation and card-account information flow for the decoupled authentication approach

6.2.3 Embedded flow

This is the 'happy path' flow for creating a consent using the embedded authentication approach.

In the embedded flow, all the authentication elements needed by the ASPSP to authenticate the PSU are exchanged through the APIs in a chain of successive requests sent by the ASPSP to the TPP, until the API Provider (ASPSP) receives all the elements. The ASPSP requests the authentication elements by sending, in the POST and PUT responses, the "_links" parameter containing sub-parameters that inform the TPP which element is required and the URL of the endpoint that the TPP should use with the PUT operation to update the authentication data in the ASPSP (e.g., "_links":{"updatePsuAuthentication":"aspsp-cde/v1/consents/consent-id"}). In addition to the "_links" parameter, the API Provider (ASPSP) may send further parameters to provide the TPP with information on the data to be requested from the PSU.

Please refer to [BG-IG] for the complete list of values for the "_links" parameter and additional parameters (e.g.: "chosenScaMethod" and "challengeData").

The authentication data required to authenticate the PSU depends on the ASPSP.

Whenever the parameter "psuMessage" is sent by the ASPSP in the response to the POST and PUT API operations, the TPP should display its content to the PSU, as these may provide the PSU with information and guidance (e.g.: informing the PSU of the User Identification they need to enter for the "updatePsuIdentification" request).

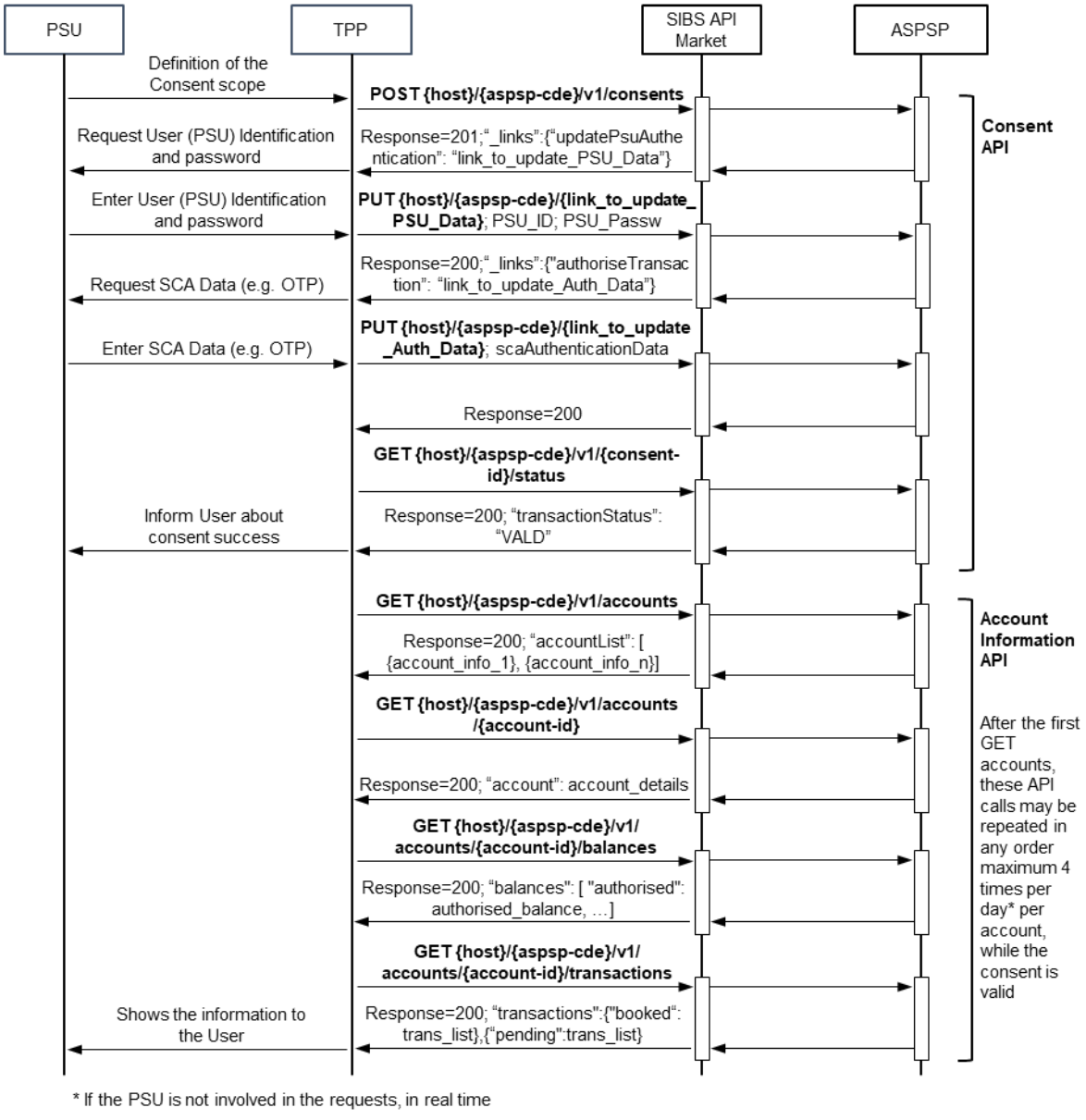


Figure 10 - Consent creation and account information flow for the embedded authentication approach

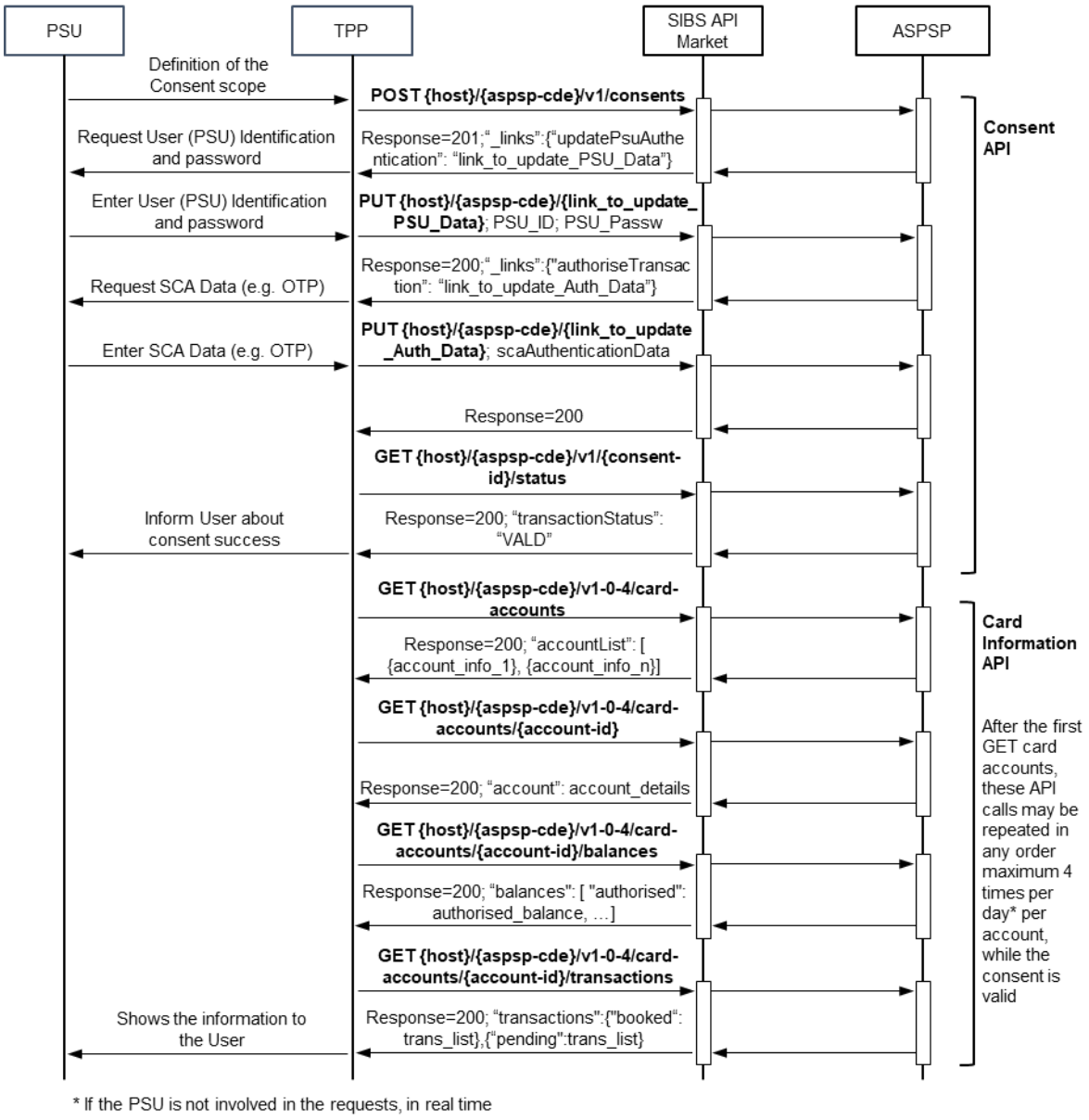


Figure 11 - Consent creation and card-account information flow for the embedded authentication approach

6.3 Authorisations

The new version of Berlin Group's NextGen PSD2 framework provides for the possibility of carrying out transactions that require the authorisation of several account holders, as is often the case with companies, exclusively through APIs. To allow several account holders to authorise a transaction and respective authentications, a new authorisation management model based on Authorisation Resources has been introduced. For each payment initiation or consent resource, one or more authorisation resources are created. The entire authorisation and authentication process is managed through these resources.

Implicit and explicit creation of Authorisation Resources

The BG specification provides for the possibility of creating authorisation resources implicitly in the POST for creating the payment initiation and consent.

- Implicit creation of the authorisation resource

Whenever the TPP does not send the TPP-Explicit-Authorisation-Preferred flag, or sends it as "false", an implicit creation of an authorisation resource is performed, regardless of whether one or multiple authentications are required. The PSU must carry out authentication immediately after the response to the POST for payment initiation creation and consent, and immediately after the DELETE for payment cancellation, where authentication is required. This applies either to one PSU only, in case one authorisation is required, or to the first of several PSUs, in case several authorisations are required.

This is the default mechanism for most ASPSPs. Please note that, in particular for consent creation, the vast majority of ASPSPs only require authorisation by one PSU and therefore prioritise the implicit creation of the authorisation resource.

- Explicit creation of the authorisation resource

Whenever the TPP sends the Authorisation-Preferred = true flag, the authorisation resource must be explicitly created by the TPP, regardless of whether it requires only one or several authentications.

When the TPP invokes an API to query the Bank's Payment/Consent Status, the status of the associated authorisations is not returned, i.e., SIBS may have consents and payments in the final status, however, the respective authorisations (or some of them) may not be in final statuses.

These authorisation statuses are updated when the TPP invokes the API to query authorisation status or when the bank updates their status.

Some ASPSPs may not offer explicit creation of the authorisation resource, especially at an early stage of release 4 implementation.

The possible values for the "transactionStatus"/code parameter and the definition of the final status are included in the following table:

Table 7 – Authorisations - Possible values for the "transactionStatus"/code parameter

"transactionStatus"	Status	Definition
RCVD	Received	An authorisation or cancellation-authorisation resource has been created successfully. This is an initial status.
SCAM	SCAMethodSelected	The PSU/TPP has selected the relevant SCA routine. If the SCA method is chosen implicitly because there is only one SCA method available, then this will be the first status reported, instead of "Received". This is an initial or intermediate status.
PSAT	PSUAuthenticated	The PSU related to the authorisation or cancellation-authorisation resource has been identified and authenticated e.g., by a password or by an access token. This is an intermediate status.

"transactionStatus"	Status	Definition
STRT	Started	The addressed SCA routine has been initiated. This is an intermediate status.
FNLS	Finalised	The SCA routine has been successfully completed (including a potential confirmation command). This is a final status.
FALD	Failed	The SCA routine has failed. This is a final status.
EXMP	Exempted	Exemption from SCA for the respective transaction, the related authorisation was successful. This is a final status.

7 FAQs

1. **How many days of transaction history can be retrieved, during the first consent request with the authorisation from the PSU?**

   There is no limit to the number of transaction history requests, but the available period depends on the ASPSP and the transaction history they make available through their digital channels.
   During the first request, after the PSU's consent is granted, the TPP has 30 minutes to retrieve as much transaction history as possible.

2. **How many days of transaction history can be retrieved via recurring transaction calls with a consent token?**

   The transaction history of the last 90 days is provided without the PSU's involvement.

3. **If a transaction ID is provided by the API, is that ID only consistent/valid within a session (access token) or does it remain valid afterwards?**

   The transaction ID is unique and remains valid afterwards.

4. **What is the maximum number of days an approved consent token can be used before expiring and needing to be refreshed or renewed?**

   Since July 25, 2023, the maximum number of days for a consent is 180. Previously, the maximum was 90 days (according to revised Article 10 of the EBA RTS).

5. **Is a recurring access to the list of accounts via a consent token supported?**

   Yes.

6. **Is a recurring access to the additional account details via a consent token supported?**

   Yes.

7. **Is a recurring access to the list of transactions via a consent token supported?**

   Yes.

8. **Is a recurring access to the balance via a consent token supported?**

   Yes.

9. **Can the PSU revoke a consent via their online banking portal?**

   Yes (status 'RevokedByPSU').

10. **How can a consent be terminated via API?**

    There are three ways to terminate a consent are: "Revoked by PSU" (via the ASPSP direct channel), "Terminated by TPP" or "Expired" (upon reaching the expiration date).

11. Can the TPP configure the lifetime of a consent?

Yes, considering the maximum duration of a consent is 180 days.

12. Can the PSU change the lifetime of a consent?

Yes, as long as it does not exceed the maximum of 180 days.

13. Can the scope of a consent be configured by the TPP?

Yes.

14. Can the PSU change the scope of a consent (e.g.: before final confirmation)?

Yes. However, change of scope after the consent creation triggers a new consent request that will replace the previous consent and consequently have a new consent ID assigned.

15. Can the TPP configure accounts for a consent?

Yes. There are two options for a consent account request: the TPP can request access to all accounts or allow the PSU to select the accounts to which they want to give access. This may depend on the ASPSP implementation.

16. Can the PSU reconfigure accounts for a consent?

Not exactly. Only when, before consent confirmation, the TPP allows the PSU to select the accounts to which they want to grant access.

17. What is the process for deploying new releases?

The standard procedure for implementing changes to the technical specifications or deploying releases of the SIBS API Market interface is to notify TPPs. This notification is accompanied by the relevant version of the User Guide and is provided at least three months prior to the scheduled implementation of the change (according to the point 4 of Article 30 of the EBA Regulatory Technical Standards (RTS) on Strong Customer Authentication and Secure Communication under PSD2).

Additionally, whenever a new release is disclosed to the TPP, assurance is provided that the developments has already been deployed in Pre-Production/Quality (QLY) environment, and the developer portal provides all the API specifications.

Exceptions are made for emergency situations, in which expedited deployment may be necessary to address critical issues. In these cases, TPPs will promptly receive notification of the emergency release and relevant deployment information.

All emergency situations in which changes are applied will be documented and made available to the proper authorities upon request.

18. How is the certificate renewal process carried out?

Renewing eIDAS certificates (QSEAL and QWAC) for TPPs on the SIBS API Market platform in the QLY and PRD environments involves the same steps as the initial registration process. TPPs seeking to renew their certificates should follow these steps:

1. Access the following link: TPPs Registration - SIBS Pay;
2. Select the option "I want to consume (Required)" and then choose "PSD2 APIs";
3. Upload the PKCS#10 files for both the QSEAL and QWAC certificates, ensuring that they adhere to the specified file type: zipped file with a maximum size of 2 MB;
4. Fill in the requested fields, paying special attention to the mandatory fields, such as "Name", "Email" and "Phone Number".

No unnecessary TPP information is required to proceed with TPP registration or certification renewal. After completing these steps, a notification from the SIBS support team will be received, confirming that the renewal process has been completed successfully.

19. Is it possible to use different API versions?

The standard procedure on the SIBS API Market platform is to perform request/response operations using the same API version for the same ASPSP. Any deviation from this standard process would depend on the ASPSP's specifications. However, it is important to note that using different API versions may not accommodate all information according to the specifications of the relevant API or the ASPSP. Therefore, it is generally recommended to adhere to the standard procedure of using the same API version for request/response operations on the SIBS API Market platform.

TPPs can use different API versions for different ASPSPs, especially if these ASPSPs are at different stages of updating or adopting the latest versions of the APIs.

For instance, a given ASPSP may have migrated to a new version of the API, while another ASPSP may still be using the previous one. In this case, TPPs interacting with these ASPSPs will need to be aware of the different API versions and adapt their integrations according to the specifications of each ASPSP, as listed in the 'List of Banks' API.

20. What is the process for deploying my application in the Production (PROD) environment?

After completing the onboarding process and thorough testing, these steps must be followed to transition the application to the Production environment (to go Live):

1. Submit a request on the developer portal: access the developer portal and submit a request to deploy the application in the Production environment. Provide all the necessary details and documentation required for the Production deployment;
2. Open a ticket: open a ticket with the proper support team to formalise the request for accessing the PROD environment. Ensure the ticket contains essential information about the app and the transition process.

Following these steps and submitting the necessary requests will trigger the transition of your application(s) into the PROD environment. This will ensure a smooth transition and proper communication with the platform's support team.

21. What are the differences between debit and credit account balances?

The way balances are displayed for debit and credit accounts may differ based on their nature and functionalities. Here are some key differences:

1. Debit Account (Checking Account):
    - Available Balance: typically displays the actual amount of funds available for withdrawal or spending;
    - Account Balance: shows the total amount of funds in the account, including pending transactions and any overdraft protection;
    - Transaction History: provides details of both incoming and outgoing transactions, reflecting the actual movement of funds in and out of the account;
    - Overdraft Limit: shows the maximum negative balance allowed before overdraft fees or penalties are incurred.
2. Credit Account (Credit Card Account):
    - Available Credit: shows the amount of credit available for spending, representing the difference between the credit limit and the current balance;
    - Current Balance: reflects the total outstanding balance owed on the credit card, including all purchases, cash advances, fees, and interest charges;
    - Minimum Payment Due: shows the minimum amount that must be paid by the due date to avoid late fees and penalties;
    - Transaction History: displays details of all transactions made using the credit card, including purchases, payments, cash advances, and fees.

Overall, while debit account balances represent available funds that can be accessed immediately, whereas credit account balances reflect outstanding debts owed to the Issuer and are subject to repayment according to the terms of the credit agreement.

22. What types of Cards does the ASPSP make available on the API Account Information?

Most ASPSPs only display Credit Cards in the API Account Information, since the API Accounts already provide similar information on Debit Card transactions and balances. Also, meal cards, prepaid cards, or other benefit cards are not displayed, since they are not considered payment accounts.

23. What types of Cards does ASPSP make available on the API Payment Initiation?

Depending on the services offered by ASPSPs on their digital channels (home banking/mobile app), the API Payment Initiation can offer payment initiation for Debit Cards, Credit Cards, or both, as well as the related subset of payment instruments available in each category. Also, meal cards, prepaid cards, or other benefit cards are not displayed, since they are not considered payment accounts.

24. What is the service Rate Limit/Burst Limit for each ASPSP?

SIBS API Market provides a set of control mechanisms to ensure stability and prevent the misuse of API calls. For the dedicated API interface, the current rate limit is 25 requests per second per Product (Accounts, Payment Initiation and MULTIBANCO Payment Initiation). Each product includes several APIs, and each API contains multiple operations. This rate limit applies at the service level, covering all requests collectively, regardless of the ASPSP, and not per each ASPSP. Once this limit is reached, further API calls are rejected with the HTTP response code 429 ('Too Many Requests').

25. How to update TPPs e-mail address?

SIBS uses the e-mail address provided by the TPP during the onboarding process to send communications and alerts regarding the service. If it is necessary to update or add an e-mail address, a ticket should be open for that prupose.

It is crucial that TPPs ensure they receive all relevant communications from SIBS, as this will keep them informed of any major changes to the service, alerts and new releases.

26. Is it possible to reopen a closed ticket?

In the Ticket Service Support, once a ticket has the status "Closed," it cannot be reopened. While it is possible to send a new message through the service, it will not be processed by the Support Team. If the ticket is closed, a new ticket should be open, even if the issue being reported is the same as the one previously addressed.