



PSD2 – APIs Implementation Guide v1.1 for TPPs

Versión: 1.8.5

Authorisation and version control

Version	Date	Concerns	Brief description of the change
1.8.4	March 2021	ALL	Adaptation to BG v1.3.8
1.8.5	July 2022	ALL	Minor bug fixes: <ul style="list-style-type: none">- Chapter 2.0: supported services review- Chapter 4: general review

TABLE OF CONTENTS

<u>1. INTRODUCTION</u>	<u>1</u>
1.1 SCOPE	1
1.2 CONTEXT	1
1.3 GLOSSARY	1
<u>2. SYSTEM OVERVIEW</u>	<u>3</u>
<u>3. TRANSPORT LAYER</u>	<u>5</u>
3.1 COMMUNICATIONS BETWEEN TPP - ASPSP	5
<u>4. APPLICATION LAYER</u>	<u>6</u>
4.1 LOCATION OF MESSAGE PARAMETERS	6
4.2 SIGNING MESSAGES UNDER BG 1.3.X SPECIFICATIONS	7
4.2.1 SIGNING MESSAGES BETWEEN TPP - ASPSP TPP -	7
4.3 API INTERFACE STRUCTURE	8
4.4 PSU CONTEXT DATA REQUIREMENTS (HTTP HEADERS)	8
4.5 REQUIREMENTS ON TPP URIs TO BE APPLIED BY THE ASPSP	11
4.6 ADDRESSING PROCESS OF THE API BY HYPERLINKS	11
<u>5. API ACCESS METHODS</u>	<u>12</u>
5.1 OAUTH2 ENDPOINTS	12
5.2 PAYMENT ENDPOINTS	13
5.3 ACCOUNT ENDPOINTS	15
5.4 TRUSTED PAYEES ENDPOINTS	16
5.5 ACCOUNT CONSENT ENDPOINTS	16
5.6 FUND CONFIRMATION CONSENT ENDPOINTS	18
5.7 FUND CONFIRMATION ENDPOINTS	19
5.8 VALUE ADDED SERVICES (VAS) ENDPOINTS	19
<u>6. DESCRIPTION OF CORE SERVICES</u>	<u>21</u>
6.1 OAUTH2 AS A PRE-STEP	21
6.1.1 FLOW	21
6.1.2 GET AUTHORIZATION	23
6.1.2.1 Request	23
6.1.2.2 OK response	26
6.1.2.3 Error response	27
6.1.2.4 Examples	27
6.1.3 GET ACCESS TOKEN	28
6.1.3.1 Request	28

PSD2 - APIs Implementation Guide v1.1 for TPPs

6.1.3.2	OK response	30
6.1.3.3	Error response	30
6.1.3.4	Examples	31
6.2	TOKEN RENEWAL	31
6.2.1	FLOW	32
6.2.2	REQUEST	33
6.2.3	RESPONSE	34
6.2.4	EXAMPLES	34
6.3	PIS: PAYMENT INITIATION SERVICE	35
6.3.1	PAYMENT INITIATION FLOWS	35
6.3.1.1	SCA flow by redirection: implicit start of authorization process	35
6.3.1.2	SCA flow by redirection: explicit start of authorization process.	38
6.3.2	PAYMENT START	44
6.3.2.1	Request	44
6.3.2.2	Response	48
6.3.2.3	Examples	50
6.3.3	FUTURE PAYMENT START	52
6.3.3.1	Request	52
6.3.3.2	Response	53
6.3.3.3	Examples	55
6.3.4	INITIATION OF STANDING ORDERS FOR RECURRING/PERIODIC PAYMENTS	56
6.3.4.1	Request	57
6.3.4.2	Response	58
6.3.4.3	Examples	59
6.3.5	GET PAYMENT STATE	60
6.3.5.1	Request	60
6.3.5.2	Response	61
6.3.5.3	Examples	62
6.3.6	RETRIEVE PAYMENT INITIATION INFORMATION	63
6.3.6.1	Request	63
6.3.6.2	Response	64
6.3.6.3	Examples	65
6.3.7	CANCEL START OF PAYMENT	67
6.3.7.1	Request	67
6.3.7.2	Response	68
6.3.7.3	Examples	69
6.4	AIS: SERVICE TO ESTABLISH CONSENT OF INFORMATION ABOUT ACCOUNTS	70
6.4.1	CHARACTERISTICS OF CONSENT	70
6.4.1.1	Consent model	70
6.4.1.2	Recurrence in access	71
6.4.1.3	Return of the account holder's name	72
6.4.1.4	List of standing orders	72
6.4.1.5	List of trusted payees	72
6.4.1.6	Consent state information	72
6.4.2	ACCOUNT INFORMATION CONSENT FLOWS	73

PSD2 - APIs Implementation Guide v1.1 for TPPs

6.4.2.1	SCA flow by redirection: implicit start of authorization process	73
6.4.2.2	SCA flow by redirection: explicit start of authorization process.	77
6.4.3	PAYMENT ACCOUNT INFORMATION CONSENT	77
6.4.3.1	Request	77
6.4.3.2	Response	82
6.4.3.3	Examples	84
6.4.4	OBTAIN CONSENT STATE	87
6.4.4.1	Request	87
6.4.4.2	Response	88
6.4.4.3	Examples	89
6.4.5	RETRIEVE CONSENT INFORMATION	90
6.4.5.1	Request	90
6.4.5.2	Response	91
6.4.5.3	Examples	93
6.4.6	REMOVE CONSENT	95
6.4.6.1	Request	95
6.4.6.2	Response	95
6.4.6.3	Examples	96
6.5	AIS: ACCOUNT DATA READING SERVICE	97
6.5.1	READING LIST OF ACCOUNTS	97
6.5.1.1	Request	98
6.5.1.2	Response	99
6.5.1.3	Examples	100
6.5.2	READING ACCOUNT DETAILS	102
6.5.2.1	Request	103
6.5.2.2	Response	104
6.5.2.3	Examples	104
6.5.3	BALANCE READING	106
6.5.3.1	Request	106
6.5.3.2	Response	107
6.5.3.3	Examples	108
6.5.4	READING OF TRANSACTIONS	109
6.5.4.1	Request	110
6.5.4.2	Response	113
6.5.4.3	Examples	113
6.6	AIS: OBTAIN LIST OF TRUSTED PAYEES	119
6.6.1	REQUEST	119
6.6.2	RESPONSE	120
6.6.3	EXAMPLES	121
6.7	FCS: ESTABLISH CONSENT FOR FUNDS CONFIRMATION SERVICE	122
6.7.1	FUND CONFIRMATION CONSENT	122
6.7.1.1	Request	122
6.7.1.2	Response	126
6.7.1.3	Examples	129
6.7.2	OBTAIN CONSENT STATE	131

PSD2 - APIs Implementation Guide v1.1 for TPPs

6.7.2.1	Request	131
6.7.2.2	Response	132
6.7.2.3	Examples	133
6.7.3	RETRIEVE CONSENT INFORMATION	134
6.7.3.1	Request	134
6.7.3.2	Response	135
6.7.3.3	Examples	136
6.7.4	REVOKE CONSENT	137
6.7.4.1	Request	137
6.7.4.2	Response	138
6.7.4.3	Examples	139
6.8	FCS: FUND CONFIRMATION SERVICE	139
6.8.1	FUND INQUIRY	139
6.8.1.1	Request	140
6.8.1.2	Response	141
6.8.1.3	Examples	142
6.9	SESSIONS: COMBINATION OF AIS AND PIS SERVICES	143
6.10	PROCESSES COMMON TO SERVICES	143
6.10.1	START THE AUTHORIZATION PROCESS (EXPLICIT)	143
6.10.1.1	Request	145
6.10.1.2	Response	148
6.10.2	GET AUTHORIZATION SUB-RESOURCES	149
6.10.2.1	Request	149
6.10.2.2	Response	151
6.10.2.3	Examples	151
6.10.3	GET SCA STATE	152
6.10.3.1	Request	152
6.10.3.2	Response	153
6.10.3.3	Examples	154
7.	DESCRIPTION SERVICES OF ADDED VALUE	156
7.1	ASPSPs SERVICE AVAILABLE	156
7.1.1	VERSION 1	156
7.1.1.1	Request	156
7.1.1.2	Response	157
7.1.1.3	Examples	157
7.1.2	VERSION 2	158
7.1.2.1	Request	158
7.1.2.2	Response	159
7.1.2.3	Examples	159
7.2	SVA: START OF PAYMENT WITH LIST OF ACCOUNTS AVAILABLE FOR PISP	160
7.2.1	PAYMENT INITIATION FLOWS	160
7.2.1.1	SCA flow by redirection with account selection: implicit start of authorization process	160
7.2.1.1	SCA flow by redirection: explicit start of authorization process	164

PSD2 - APIs Implementation Guide v1.1 for TPPs

7.2.2	PAYMENT INITIATION COMPLETION	164
7.2.2.1	Request	164
7.2.2.2	Response	165
7.2.2.3	Examples	165
7.3	SVA: START OF STANDING ORDERS FOR RECURRING / PERIODIC PAYMENTS WITH LIST OF ACCOUNTS AVAILABLE FOR PISP	166
7.3.1	PERIODIC PAYMENT INITIATION FLOWS	167
7.3.1.1	SCA flow by redirection with account selection: implicit start of authorization process	167
7.3.1.2	SCA flow by redirection: explicit start of authorization process	172
7.3.2	PAYMENT INITIATION COMPLETION	172
7.3.2.1	Request	173
7.3.2.2	Response	175
7.3.2.3	Examples	178
8.	DEFINITION OF TYPES OF COMPOSITE DATA	180
8.1	ACCOUNTACCESS	180
8.2	ACCOUNTDETAILS	182
8.3	ACCOUNTREFERENCE	184
8.4	ACCOUNTREPORT	185
8.5	ADDITIONALINFORMATIONACCESS	186
8.6	ADDRESS	186
8.7	AMOUNT	187
8.8	AUTHENTICATIONOBJECT	187
8.9	ASPSP	188
8.10	BALANCE	189
8.11	EXCHANGERATE	189
8.12	HREF	190
8.13	LINKS	190
8.14	SINGLEPAYMENT	193
8.15	STANDINGORDERDETAILS	194
8.16	STRUCTUREDADDITIONALINFORMATION	197
8.17	TPPMESSAGE	197
8.18	TRANSACTIONS	197
8.19	TRUSTEDBENEFICIARY	202
9.	ANNEXES	202
9.1	SIGNATURE	202
9.1.1	HEADER "DIGEST" REQUIRED	203
9.1.2	SIGNATURE REQUIREMENTS	203
9.1.3	EXAMPLE	205
9.1.3.1	Generation of the header "Digest"	206
9.1.3.2	Generation of the header "Signature"	206
9.1.3.3	Generation of the header "TPP-Signature-Certificate"	207

PSD2 - APIs Implementation Guide v1.1 for TPPs

9.1.3.4	Definitive headers to send	207
9.2	HTTP RESPONSE CODES	208
9.3	RETURN CODES	209
9.4	TRANSACTION STATES	214
9.5	CONSENT STATES	215
9.6	TYPES OF BALANCES	216
9.7	TYPES OF COMMISSION SHARING	216
9.8	SCA STATES	216
FAQ: FREQUENTLY ASKED QUESTIONS		217

1. INTRODUCTION

1.1 Scope

This document corresponds to the Technical Design of the interface between payment service providers (TPPs) and HUB for compliance with the PSD2 directive.

1.2 Context

Final document between Redsys and Financial Entities associated with the HUB.

1.3 Glossary

The following table lists the acronyms and definitions used throughout the document.

Abbreviation	Definition
ASPSP	Payment service provider account manager
	Provides and maintains client accounts from which payments can be made.
PISP	Payment initiation service provider
	initiates a payment order, at the user's request, from a payment account of another provider
AISP	Account information service provider
	Provide the client with information about his/her payment accounts with other providers.
TPP	Third party provider

PSD2 - APIs Implementation Guide v1.1 for TPPs

	runs the services defined by PSD2 on behalf of a PSU. If required for service, access the account (s) of the PSU managed by an ASPSP using the XS2A Interface of that ASPSP. It sends request messages to the XS2A interface of the ASPSP and receives corresponding response messages from that ASPSP.
PIISP	Payment service provider issuing payment instruments
	Provides the user with a payment instrument with which to initiate and process payment transactions.
PSU	It can be a natural or legal person following PSD2 legislation. It instructs the TPP implicitly or explicitly to perform any PSD2 service towards its ASPSP.
MA	Mandatory
COND	Conditional
OP	Optional

2. SYSTEM OVERVIEW

The following lists the available services:

Service		Function	State
CORE	PIS	Start of simple single-signature payment	Available
		Start of recurring payments	Available
		Initiation of multiple payments/bulk	Not Supported
		Start of future payments	Available
		Check Payments State	Available
		Retrieve payment initiation information	Available
		Execute payment start	Available
		Payment Cancellation	Available
	AIS	Consent of information about payment accounts	Available
		Consent of information about card accounts	Not Supported
		Retrieve consents information	Available
		Check consent state	Available
		Remove consent	Available
		Reading list of accounts available with / without balances	Available
		Reading list of accessible accounts with / without balances	Available
		Reading account details with / without balances	Available
		Balance reading	Available
		Read transactions with / without balances	Available
		Reading of transaction detail	Not Supported
		List of standing orders	Available
		Trusted payees	Available
		Reading card account list	Not Supported
		Reading card account details	Not Supported
		Reading of card account balances	Not Supported
		Reading card account transactions	Not Supported
	FCS	Establish consent	Available
		Retrieve consent information	Available
		Check consent state	Available
		Remove consent	Available

PSD2 - APIs Implementation Guide v1.1 for TPPs

	SCA	Confirmation of funds	Available
		SCA per flow redirect	Available
		Uncoupled flow SCA	Not Supported
		SCA embedded	Not Supported
	Common processes	Start explicit authorization	Available
		SCA state query	Available
		Get authorization sub-resources	Available
		Update authorization data	Not Supported
	OAUTH	Obtaining access token	Available
		Access token renewal	Available

Table 1: CORE Services

Service		Function	State
SVA	PIS	Start of payment with list of accounts available for PISP	Available
		Start of recurring payments with list of accounts available for PISP	Available
	AIS	Notice of data available in PUSH mode	Not Supported
	DIR. TPPs	List of available TPPs	Not Supported
		TPP information query	Not Supported
		New TPP notification	Not Supported

Table 2: Value Added Services

3. TRANSPORT LAYER

The following information is valid for PRODUCTION environment.

3.1 Communications between TPP - ASPSP

Channel https (TLS 1.2) + TWOWAY-SSL

The communication between the TPP and the ASPSP is always secured by the use of a MATLS 1.2 connection (2WAYSSL) with client authentication.

In summary, the validations to apply:

- TLS 1.2 communication with Mutual Authentication with Client Certificate (MATLS 1.2)
- Based on X509 certificates from recognized CAs (Digicert) and eIDAS certificates from TPPs issued by valid QTSPs PSD2
- Temporary validity of the certificate
- Common Name of the certificate Subject should be the one expected
- Complete certification chain validation
- CRL validation

4. APPLICATION LAYER

4.1 Location of message parameters

The definition of the interface follows the REST services approach. This approach allows message parameters to be transported at different levels:

- Message parameters as part of the HTTP layer (HTTP headers)
- Message parameters defining additional query parameters in the path (information in the path of the URL)
- Message parameters as part of the HTTP body

The parameters contained in the corresponding HTTP body will be encoded in JSON.

The parameters are encoded in:

- spinal-case (lowercase letters) at path level
- Spinal-case (starting with capital letters) at the HTTP header level
- lowerCamelCase for query parameters (query params) and JSON-based parameters.

The following principles apply in the definition of the API:

- Defining the content syntax
- Certificates and signature data required
- PSU identification data (based on access token)
- Protocol level data such as request timestamp or request/transaction identifiers

Message parameters as part of the path level:

- Provider identification
- Service identification
- Payment Type Identification
- Resource ID

Query parameters:

- Additional information needed to process GET requests to filter information

Message parameters as part of the HTTP body:

- Business data
- PSU authentication data
- Information Messages
- Hyperlinks to fully address the TPP-ASPSP process

4.2 Signing messages under BG 1.3.x specifications

All requests will be signed to the ASPSP.

4.2.1 Signing Messages Between Tpp - Aspsp Tpp -

The TPP will always sign all petitions sent to the ASPSP and the ASPSP must validate them.

The signature must be included in the HTTP headers as defined in the Berlin Group - Implementation Guidelines, chapter 4.

The electronic signature of the TPP is based on a certificate for electronic signature. This certificate must be issued by a valid QTSP PSD2.

In summary, the validations to apply:

- Based on eIDAS issued by valid QTSP PSD2
- Temporary validity of the certificate
- Common Name of the certificate Subject should be the one expected
- Complete certification chain validation
- CRL validation
- Signing of the message following the Berlin Group standard - Implementation Guidelines v1.3.x

In general, all requests (except for OAuth2 authorize as a pre-step) will include the following header fields for the message signature:

Field	Description	Type	Mand at.	Format
Digest	It is contained if the Signature field is travelling. See 9.1 Signature for more information. tpp documentation.	String	MA	^.{1,100} \$ Ex: Digest: SHA-256=NzdmZjA4YjY5M2M2NDYyMmVjOWFmMGNmYTZiNTU3MjVmNDI4NTRIMzJkYzE3ZmNmMDE3ZGFmMjhhNTc5OTU3OQ==
Signature	Signing of the request by the HUB. See 9.1Signature for more information.	String	MA	See 9.1Signature for more information.

TPP-Signature-Certificate	HUB certificate used to sign the Base64 request. See 9.1Signature for more information.	String	MA	^.{1,5000}\$ EX: TPP-Signature-Certificate: MIIHgZCCBmugAwIBA gIIZzZvBQlt0UcwDQYJKoZIhvcNAQEL BQAwSTELMAkGA1UE BhMCMVVMxEzARBgNVB A
----------------------------------	--	--------	----	--

4.3 API interface structure

The interface is resource oriented. Resources can be directed under the API endpoints.

Using additional content parameters {parameters}, where:

- {provider} is the host and path of the API
- v1.1 is the version of this specification
- {service} has the values consents, Payments, bulk-payments, periodic-payments, accounts, card-accounts, or funds-confirmations, and which are extended by adding more information related to the type of product and the scope requested.
- {query-parameters} are parameters that provide details about GET access methods
- {parameters} are attributes defined in JSON encoding

The structure of the request / response is described according to the following categories:

- Path: attributes encoded in the Path
- Query parameters: attributes added to the path after the sign '?' as flags to address processes or filter attributes for GET access methods. Boolean type access parameters must always be used with the values true or false.
- Header: attributes encoded in the HTTP header of the request or the response
- Request: attributes of the request
- Response: response attributes in JSON

The HTTP response codes, which can be used in the interface, will be defined later.

4.4 PSU Context Data Requirements (HTTP headers)

PSD2 - APIs Implementation Guide v1.1 for TPPs

The following elements are used to send information about the PSU-TPP interface and are used for the ASPSP risk management procedures. It is highly recommended to send these elements in all requests for Initiate Payment or Establish Consent transaction flows. For example, in flows where a PSU authentication is required (Except in OAuth2 as a pre-step). The following table will not be repeated in the following sections for better readability. The only exception is in certain requests where a condition other than "optional" applies. For example, PSU-IP-Address.

Note: the information about the PSU-TPP interface could be used by the ASPSP as input for fraud detection and risk management systems. You can use this information also to exclude some authentication methods (for example, some ASPSP does not allow to receive an OTP by SMS on the same device that triggers the transaction). In addition, it allows ASPSPs to receive specific information from the partner device in order to be able to support an app-to-app redirection procedure for the TPP. For these reasons, it is highly recommended that TPPs include all of this information in related requests. Failure to provide all the necessary information could lead to a classification of the PSU device as unusable for the authentication method or a classification of the current transaction as "high risk", for example due to session attacks. Due to this, the probability of a rejection of the transaction due to fraud detection and / or risk management could be increased.

Field	Description	Type	Mand at.	Format
PSU-IP-Address	IP address of the HTTP request between the PSU and the TPP.	String	OP	IPv4 and IPv6 Ex: PSU-IP-Address: 192.168.16.5
PSU-IP-Port	IP port of the HTTP request between the PSU and the TPP if available.	String	OP	^.{1,5}\$ Ex: PSU-IP-Port: 443
PSU-Accept	Accept header of the HTTP request between the PSU and the TPP.	String	OP	^.{1,50}\$ Ex: PSU-Accept: application / json
PSU-Accept-Charset	Accept charset header of the HTTP request between PSU and the TPP.	String	OP	^.{1,50}\$ Ex: PSU-Accept-Charset: utf-8
PSU-Accept-Encoding	Accept encoding header of the HTTP request between PSU and the TPP.	String	OP	^.{1,50}\$ Ex: PSU-Accept-Encoding: gzip

PSD2 - APIs Implementation Guide v1.1 for TPPs

PSU-Accept-Language	Accept language header of the HTTP request between PSU and the TPP.	String	OP	$\wedge.\{1,50\}\$$ Ex: PSU-Accept-Language: es-ES
PSU-User-Agent	Browser or operating system of the HTTP request between the PSU and the TPP.	String	OP	Ex: PSU-User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.1.5) Gecko/20091102 Firefox/3.5.5 (.NET CLR 3.5.30729)
PSU-Http-Method	HTTP method used in the interface between PSU and TPP. Allowed values: <ul style="list-style-type: none"> • POST • GET • PUT • PATCH • DELETE 	String	OP	Ex: PSU-Http-Method: POST
PSU-Device-ID	UUID (Universally Unique Identifier) for a device. The UUID identifies the device or an installation of an application on a device. This ID should not be modified until the application is uninstalled from the device.	String	OP	UUID $\wedge[0-9a-fA-F]\{8\}-[0-9a-fA-F]\{4\}-[0-9a-fA-F]\{4\}-[0-9a-fA-F]\{4\}-[0-9a-fA-F]\{12\}\$$ Ex: PSU-Device-ID: 5b3ab8e8-0fd5-43d2-946e-d75958b172e7
PSU-Geo-Location	Location corresponding to the HTTP request between the PSU and the TPP	String	OP	RFC 2426 $\wedge\text{GEO}:[\\d]*.[\\d]*[; ,][\\d]*.[\\d]*\$$ Ex: PSU-Geo-Location: GEO:90.023856;25.345963

4.5 Requirements on TPP URIs to be applied by the ASPSP

The TPP can provide multiple URIs to the ASPSP as parameters for the next steps of the protocol.

For security reasons, it must be ensured that these URIs are secured by the certificate used by the TPP for their identification. Apply as follows:

The URIs provided by the Hub in the TPP-Redirect-URI or TPP-Nok-Redirect-URI fields must comply with the domain secured by the TPP certificate in the CN field or in its SubjectAltName. It is taken into account that for cases like example-hub.com in the TPP-Redirect-URI like:

- www.example-hub.com/xs2a/v1.1/service/asdf o
- redirections.example-hub.com/xs2a/v1.1/service/asdf

They would be valid cases.

Wildcard certificates are taken into account to validate.

Requests that do not meet the requirement will be rejected.

4.6 Addressing process of the API by hyperlinks

The XS2A API requires several requests for the initiation of payment and account information services from the TPP to the ASPSP. In Initiate Payment requests and Establish Consent requests, a resource is generated by the ASPSP. The "location" header of the response will normally contain a link to the created resource.

Additionally, the ASPSP can embed the hyperlink together with a "tag" for the semantics of the same in the response of these first requests and for all the following requests in the services. This hyperlink will be relative to save space, except in cases such as redirections where it will be absolute.

The hyperlink "tag" carries the functionality of the resource directed by the link. For example, "authorise-transaction". This link indicates that the results of the SCA method must be sent to the resource directed by this link to authorize, for example, a payment.

The hyperlinks for addressing are carried in the "_links" element. This can contain one or more hyperlinks.

5. API ACCESS METHODS

The following tables provide an overview of the HTTP access methods supported by API endpoints and API-generated resources.

Conditions in the following tables

Additionally, it is defined when a supported method is mandatory for ASPSP by this specification or when it is an optional feature. It should be noted that the given condition is relative to the parent node of the path. For example, the condition on the GET method `/v1.1/consents/{consentId}` applies only if the POST endpoint `/v1.1/consents` is supported.

It should be noted that any of the methods used by the TPP, which are addressing dynamically created resources in this API, can only apply to resources which have been created before by the TPP itself.

5.1 OAuth2 endpoints

Endpoint	Method	Cond.	Description
/authorize	GET	MA	Redirection to the ASPSP login website to obtain the authCode.
/token	POST	MA	Allows to send the authCode to obtain the access token.
/token	POST	MA	Refresh the access token if it has expired.

5.2 Payment Endpoints

Endpoint	Method	Cond.	Description
/payments/{payment-product}	POST	MA	Creates a payment initiation resource accessible under the {paymentId} with all relevant data for the corresponding payment product. This is the first step in the API to initiate the referred payment.
/payments/{payment-product} /{paymentId}	GET	MA	Gets the details of an initiated payment.
/payments/{payment-product} /{paymentId}/state	GET	MA	Gets the state of the payment transaction.
/periodic-payments/{payment-product}	POST	OP	Creates a standing order resource for periodic/recurring payment accessible under the {paymentId} with all relevant data for the corresponding payment product. This is the first step in the API to initiate the referred periodic/recurring payment.
/payments /{payment-product} /{paymentId}	GET	MA	Gets the details of an initiated standing order for periodic/recurrent payment.
/periodic-payments/{payment-product} /{paymentId}/state	GET	MA	Gets the state of the standing order transaction for periodic/recurrent payment.
{payment-service}/{payment-product} /{paymentId}/authorisations	POST	MA	Create an authorization sub-resource and start the authorization process.

			The ASPSP could make the use of this access method unnecessary in case only a single SCA process is needed, as the related authorisation resource could be automatically created by the ASPSP after the payment data is sent with the first POST request to /payments/{payment-product}.
{payment-service}/{payment-product}/{paymentId}/authorisations	GET	MA	Gets the list of authorization sub-resource IDs that have been created.
{payment-service}/{payment-product}/{paymentId}/authorisations/{authorisationId}	GET	MA	Gets the SCA state of the authorization.
{payment-service}/{payment-product}/{paymentId}	DELETE	OP	<p>Cancel the accessible payment under the paymentId resource if applicable for the payment service, payment product and received in the period of time that the cancellation is allowed.</p> <p>The response to this DELETE command will tell the TPP when:</p> <ul style="list-style-type: none"> • The access method was rejected • The access method was correct • The access method is generally applicable but requires an additional authorization process.

5.3 Account Endpoints

Endpoint	Method	Cond.	Description
/accounts	GET	MA	<p>Reads all account identifiers for which the PSU has granted access on the /consents endpoint. In addition, relevant account information and links to the corresponding account information resources may be provided if the necessary permissions have been provided.</p> <p>Note: the endpoint /consents optionally offer to grant access on all available PSU payment accounts .In this case, this endpoint will release the information of all available payment accounts from the PSU to the ASPSP.</p>
/accounts?withBalance	GET	MA	Obtain the identifiers of the available payment accounts along with balance information, depending on the consent granted.
/accounts/{account-id}	GET	MA	Gets detailed information about the accessed account.
/accounts/{account-id}?withBalance	GET	MA	Gets detailed information on the accessed account along with balance information.
/accounts/{account-id}/balances	GET	MA	Obtains detailed information on the balances of the account accessed.

/ accounts / {account-id} / transactions	GET	MA	Obtains a list of movements (transactions) of the accessed account. For a given account, additional parameters are, for example, date from, date to, and date to.
---	-----	----	--

Note: the {account-id} parameter can be tokenized by the ASPSP in such a way that the current account numbers, such as IBANs or PANs, are not part of the API path definition for data protection reasons. This tokenization is managed by the ASPSP.

5.4 Trusted Payees Endpoints

Endpoint	Method	Cond.	Description
/trusted-beneficiaries?{account-id}	GET	OP	Obtain a list of trusted payees.

5.5 Account Consent Endpoints

Endpoint	Method	Cond.	Description
/consents	POST	MA	Creates a consent resource, defining access permissions on specific accounts of a PSU. These accounts are explicitly routable on the PATH as parameters.
/consents	POST	MA	Optionally, an ASPSP could accept specific access permissions to access all PSD2 services on all available accounts.

			<p>Another option is that an ASPSP could accept a request where only the access permissions are reported but not the accounts. In this case, the selection of accounts is subsequently managed between the PSU and the ASPSP.</p> <p>As a last option, the ASPSP can accept requests with the following access permissions:</p> <ul style="list-style-type: none"> • Get a list of available payment accounts • Get a list of available payment accounts with balances
/consents/{consentId}	GET	MA	Gets the exact definition of the consent resource, including the validity state.
/consents/{consentId}	DELETE	MA	Ends the directed consent.
/consents/{consentId}/state	GET	MA	Gets the state of the directed consent.
/consents/{consentId}/authorizations	POST	MA	Create an authorization sub-resource and start the authorization process.

			The ASPSP could make the use of this access method unnecessary in case only a single SCA process is needed, as the related authorisation resource could be automatically created by the ASPSP after the consents data is sent with the first POST /consents request.
/consents/{consentId}/authorisations	GET	MA	Gets the list of authorization sub-resource IDs that have been created.
/consents/{consentId}/authorisations/{authorisationId}	GET	MA	Gets the SCA state of the authorization.

5.6 Fund confirmation Consent endpoints

Endpoint	Method	Cond.	Description
/consents/confirmation-of-funds	POST	MA	Create a consent resource for funding confirmation.
/consents/confirmation-of-funds/{consentId}	GET	MA	Gets the exact definition of the consent resource, including the validity state.
/consents/confirmation-of-funds/{consentId}	DELETE	MA	Ends the directed consent.
/consents/confirmation-of-funds/{consentId}/state	GET	MA	Gets the state of the directed consent.
/consents/confirmation-of-funds/{consentId}/authorisations	POST	MA	Create an authorization sub-resource and start the authorization process.

			The ASPSP could make the use of this access method unnecessary in case only a single SCA process is needed, as the related authorisation resource could be automatically created by the ASPSP after the submission of the consent data with the first POST /consents/confirmation-of-funds request.
/consents/confirmation-of-funds/{consentId}/authorisations	GET	MA	Gets the list of authorization sub-resource IDs that have been created.
/consents/confirmation-of-funds/{consentId}/authorisations/{authorisationId}	GET	MA	Gets the SCA state of the authorization.

5.7 Fund Confirmation Endpoints

Endpoint	Method	Cond.	Description
/ funds-confirmations	POST	MA	Checks when a specific amount is available at a certain point in time for an account related to a TPP/card or targeted by TPP and IBAN.

5.8 Value Added Services (VAS) Endpoints

Endpoint	Method	Cond.	Description
----------	--------	-------	-------------

/sva/payments/{payment-product}	POST	MA	Creates a payment initiation resource accessible under the {paymentId} with all relevant data for the corresponding payment product. This is the first step in the API to initiate the referred payment without having to inform the issuer's account.
/sva/periodic-payments/{payment-product}	POST	MA	Creates a periodic payment initiation resource accessible under the {paymentId} with all relevant data for the corresponding payment product. This is the first step in the API to initiate the referenced periodic payment without the need to inform the issuer's account.

6. DESCRIPTION OF CORE SERVICES

6.1 OAuth2 as a pre-step

6.1.1 Flow

In the scenario shown in the figure, only OAuth2 Figure 1: OAuth2 scenario as a pre-step is represented with its "Authorisation Code Grant" flow and the involvement of the parts.

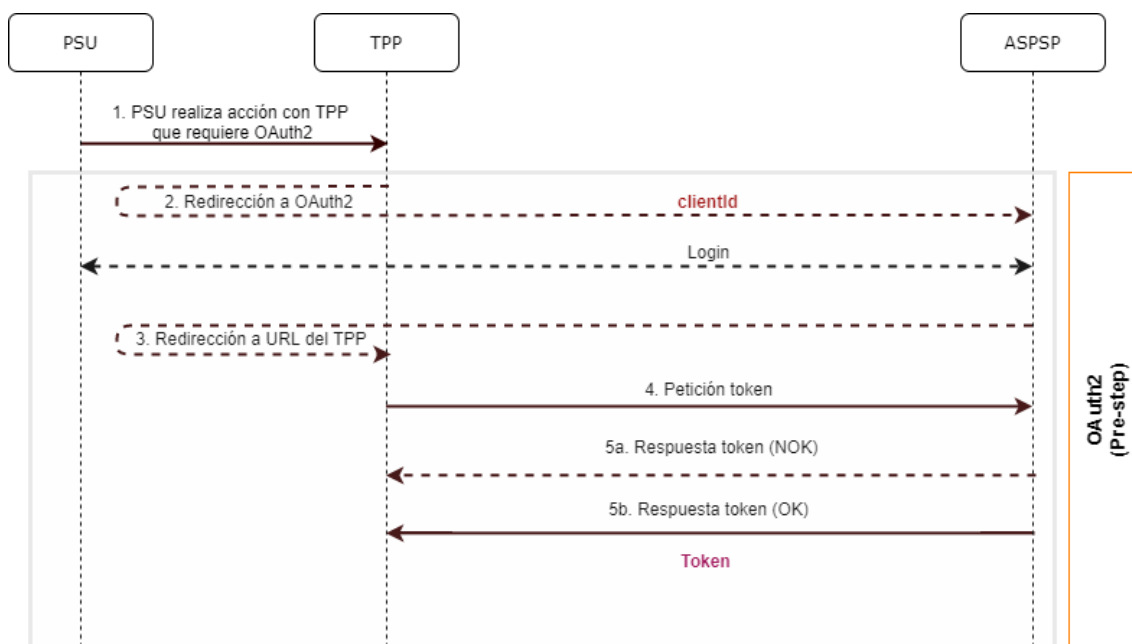


Figure 1: OAuth2 scenario as a pre-step

The points indicated in the flow are described below:

1. PSU performs action with TPP that requires OAuth2

PSU executes action that requires OAuth2.

2. Redirection to OAuth2

The TPP detects that the PSU that is trying to perform the action has not yet logged into the system and, therefore, does not have a valid access token to consume the resources.

In this situation, the TPP routes the PSU browser to perform a redirect to the authorisation URL of the OAuth2 server by informing, among other values, the *clientIdTPP* and the *redirect_uri* of the TPP to which the *callback* will be performed.

```
GET / authorize?response_type=code&client_id=PSDES-BDE-3DFD21
&state=asd&redirect_uri=https%3A%2F%2Ftp%2Eexample%2Ecom%2Fcb&code_c
hallenge=E9Melhoa2OwvFrEMTJguCHaoeK1t8URWbuGJSstw-
cM&code_challenge_method=S256
```

```
HTTP/1.1
Host: hub.example.com/aspsp-name
```

Login

The PSU enters its credentials, *user / pass*, on the *login* page of its ASPSP.

3. Redirection to TPP URL

Once the *login* has been successfully completed, the ASPSP instructs the PSU browser to perform the *callback* to the URL informed of the initial redirection (point 3 of the flow).

In this return redirection, the ASPSP, in case of successful *login*, reports an authorization code, *authCode*, which will be used in a subsequent request to request the access token from the ASPSP.

```
HTTP/1.1 302 Found
Location:
https://hub.example.com/cb?code=Sp1xl0BeZQQYbYS6WxSbIA&state=xyz
```

4. Token request (TPP ASPSP)

The TPP makes a POST request to the OAuth2 server to obtain the access token that will allow it to consume the displayed API services.

Among the possible values to send, the TPP informs its *clientId* and the *authCode* returned in the redirection (point 3).

5. Token response (ASPSP TPP)

The ASPSP evaluates the data provided in the token request and, if everything was correct, generates an access token (*token*) that will return it in the response.

Note: those ASPSPs that require a payment execution request, may also return an additional token (*authToken*) at this point that will be used later to authorize the payment.

PSD2 - APIs Implementation Guide v1.1 for TPPs

```
HTTP / 1.1 200 OK
Content-Type: application/json; charset=utf-8
Cache-Control: no-store
Pragma: no-cache
{
  "access_token": "1zCsicMWpAA2YotnFZFEjr",
  "token_type": "Bearer",
  "expires_in": 3600,
  "refresh_token": "G5Qx2TlKWIAAtGzv3JOkF0X"
}
```

6.1.2 Get authorization

6.1.2.1 Request

The TPP redirects the PSU to make one of the following requests to the Hub:

- Login via web redirect
- Login using biometric authentication by app-to-app redirection

Endpoint for web authentication

GET

`/aspsp/authorize?response_type={response_type}&client_id={client_id}&scope={scope}&state={state}&redirect_uri={redirect_uri}&code_challenge={code_challenge}&code_challenge_method={code_challenge_method}`

Endpoint for biometric authentication by app-to-app redirection for individuals

GET

`/aspsp/biometric/app-to-app/personal/authorize?response_type={response_type}&client_id={client_id}&scope={scope}&state={state}&redirect_uri={redirect_uri}&code_challenge={code_challenge}&code_challenge_method={code_challenge_method}`

Path

Field	Description	Type	Mandat.	Format
provider	URL of the ASPSP where the service is published.	String	MA	Ex: aspsp.example.es

Query parameters:

Field	Description	Type	Mand at.	Format
response_type	The value must be set to "code".	String	MA	Ex: response_type = code
client_id	<p>"organizationIdentifier" provided in the eIDAS certificate formed as:</p> <ul style="list-style-type: none"> - PSD - 2 characters of the EQS country code according to ISO 3166 - Character "-" - 2-8 characters for NCA identifier (AZ in uppercase) - Character "-" - PSP identifier <p>This registration number will be that of the HUB or the TPP depending on the ASPSP configuration.</p>	String	MA	<p>^.{1,70}\$</p> <p>Ex: client_id=PSDES-RDS-4000</p>
scope	<p>Scope possible:</p> <ul style="list-style-type: none"> • PIS • AIS • SVA <p>You can specify more than one by separating it by a space (% 20).</p>	String	MA	<p>^.{1,64}\$</p> <p>Ex: scope=PIS%20AIS%20SVA</p>
state	Opaque value generated by the TPP. Used to prevent cross-site request	String	MA	<p>^.{1,64}\$</p> <p>Ex: state = XYZ</p>

PSD2 - APIs Implementation Guide v1.1 for TPPs

	forgery XSRF attacks.			
redirect_uri	URL back to the HUB where the authorization code "code" that will be used later to obtain the access token will be reported.	String	MA	<code>^.{1,250}\$</code> Ex: <code>redirect_uri=https%3A%2F%2Fwww%2Ehub%2Ecom%2Fcb</code>
code_challenge	PKCE challenge used to prevent code injection attacks. According to RFC 7636.	String	MA	<code>^.{1,128}\$</code> Ex: <code>code_challenge=E9MeI hoa2OwvFrEMTJguCHa oeK1t8URWbuGJSstw- cM</code>
code_challenge_method	Method to verify the code that can be "plain" or "S256". Preferred S256 (SHA 256)	String	OP	<code>^.{1,120}\$</code> Ex: <code>code_challenge_method = S256</code>
second_client_id	It will receive the value of the clientId from the HUB or the TPP depending on the value of the clientId attribute depending on the ASPSP configuration.	String	OP	<code>^.{1,70}\$</code> Ex: <code>second_client_id=PSD ES-BDE-3DFD246</code>
app_to_app_preferred	Indicates whether the TPP has used the biometric authentication endpoint to receive a deeplink for app-to-app redirection. Possible values: <ul style="list-style-type: none"> • staff 	String	OP	Ex: <code>app_to_app_preferred =true</code>

Header

No additional fields are specified.

Body

No data travels in the body of this response.

6.1.2.2 OK response

Response in case the request has passed correctly. It results from the redirection initiated by the ASPSP from the PSU browser to the return URL provided by the HUB.

Path

No additional fields are specified.

Query parameters:

Field	Description	Type	Mandat.	Format
Location	Contains the URI where the redirect to the HUB is performed.	String	MA	Ex: Location: https://hub.example.es/cb
code	One-time authorization code generated by the HUB. Recommended lifetime of no more than 10 minutes.	String	MA	[A-Za-z0-9]{32} Ex: code=SpIxlOBeZQ QYbYS6WxSbIA
state	Opaque value generated by the TPP. Used to maintain state between request and response. The ASPSP will include this when redirecting the PSU browser back to the HUB. Used to prevent cross-site request forgery attacks.	String	MA	^{1,64}\$ Ex: state = XYZ

Body

No data travels in the body of this request.

6.1.2.3 Error response

Response in case an error has occurred in the request. It results from the redirection initiated by the ASPSP from the PSU browser to the return URL provided by the HUB.

Path

No additional fields are specified.

Query parameters:

Field	Description	Type	Mand at.	Format
Location	Contains the URI where the redirection to the HUB takes place	String	MA	Ex: Location: https://hub.example.es/cb
error	Code indicating the error that occurred.	String	MA	Ex: error = invalid_request
state	Value generated by the TPP. Used to maintain state between request and response. The HUB will send it back in the reply.	String	MA	^. {1,64}\$ Ex: state = XYZ

Body

No data travels in the body of this request.

6.1.2.4 Examples

Example of request

GET
https://hub.example.es/authorize?response_type=code&client_id=PSDES-RDS-4000&scope=PIS%20AIS%20SVA&state=xyz&redirect_uri=https%3A%2F%2Fwww%2Ehub%2Ecom%2Fcb&code_challenge=E9Melhoa2OwvFrEMTJguCHaoeK1t8URWbuGJSstw-cM&code_challenge_method=S256&second_client_id=PSDES-BDE-3DFD246

Example of OK response:

HTTP/1.1 302 Found

Location:

<https://hub.example.es/cb?code=Sp1xl0BeZQQYbYS6WxSbIA&state=xyz>

Example of NOK response:

HTTP/1.1 302 Found

Location: https://hub.example.es/cb?error=access_denied&state=xyz

6.1.3 Get access token

This message is sent by the TPP to the ASPSP to exchange the authorization code obtained in the previous step and obtain an access token and refresh token.

6.1.3.1 Request

Endpoint

POST {provider}/token

Path

Field	Description	Type	Mand at.	Format
provider	URL of the HUB where the service is released.	String	MA	Ex: hub.example.es

Request Parameters

Field	Description	Type	Mand at.	Format
grant_type	It must take the value of "authorization_code"	String	MA	Ex: grant_type=authorization_code
client_id	"organizationIdentifier" provided in the eIDAS certificate formed as: <ul style="list-style-type: none">- PSD	String	MA	^. {1,70}\$ Ex: client_id=PSDES-RDS-4000

PSD2 - APIs Implementation Guide v1.1 for TPPs

	<ul style="list-style-type: none"> - 2 characters of the EQS country code according to ISO 3166 - Character "-" - 2-8 characters for NCA identifier (AZ in uppercase) - Character "-" - PSP identifier 			
code	Authorisation code returned by the ASPSP in the previous request for an authorisation code application	String	MA	$\wedge.\{1,64\}\$$ Ex: code = SplxlOBeZQQY bYS6WxSbIA
redirect_uri	Exact URL of the TPP where the OAuth2 server redirected the user agent for this particular transaction	String	MA	$\wedge.\{1,250\}\$$ Ex: redirect_uri=https%3A%2F%2Fwww%2Ehub%2Ecom%2Fcb
code_verifier	PKCE verification code used to prevent code injection attacks. Based on RFC 7636.	String	MA	Ex: code_verifier=dBjftJeZ4CVP-mB92K27uhbUJU1p1r_wW1gFWFOEjXk

Header

No additional fields are specified.

Body

No fields travel in the Body.

6.1.3.2 OK response

Response in case the request has passed correctly. It is given as a result of the request to obtain access token sent by the TPP to the ASPSP.

Body

Field	Description	Type	Mand at.	Format
access_token:	Access token issued by the ASPSP and linked to the scope requested in the submission and confirmed by the PSU.	String	MA	^.{1,64}\$ Ex: "access_token":"2YotnFZFEjr1zCsicMWpAA"
token_type	Type of the issued token. It will take the value "Bearer".	String	MA	Ex: "token_type": "Bearer"
expires_in	Access token lifetime in seconds.	Integer	OP	Ex: "expires_in": 300
refresh_token	Refresh token. It can be used to obtain a new access token if it has expired.	String	OP	^.{1,64}\$ Ex: "refresh_token": "tGzv3JOkF0XG5Qx2TlKWIA"

6.1.3.3 Error response

Response in case an error has occurred in the request. It is given as a result of the access token request made by the TPP to the HUB.

Body

Field	Description	Type	Mand at.	Format
error	Code indicating the error that occurred. See more return codes in the annexes.	String	MA	Ex: "error":"invalid_request"

6.1.3.4 Examples

Example of request

```
POST /token HTTP/1.1
Host: https://aspsp.example.es
Content-Type: application/x-www-form-urlencoded
grant_type=authorization_code&client_id=PSDES-RDS-
4000&code=Sp1xl0BeZQQYbYS6WxSbIA&redirect_uri=https%3A%2F%2Fwww%2Etp%
2Ecom%2Fcb&code_verifier=dBjftJeZ4CVP-mB92K27uhbUJU1plr_wW1gFWFOEjXk
```

Example of OK response:

```
HTTP / 1.1 200 OK
Content-Type: application/json; charset=utf-8
Cache-Control: no-store
Pragma: no-cache
{
  "access_token": "2YotnFZFEjrlzCsicMWpAA",
  "token_type": "Bearer",
  "expires_in": 3600,
  "refresh_token": "tGzv3JOkF0XG5Qx2TlKWIA"
}
```

Example of NOK response:

```
HTTP / 1.1 400 Bad Request
Content-Type: application/json; charset=utf-8
Cache-Control: no-store
Pragma: no-cache
{
  "error": "invalid_request"
}
```

6.2 Token renewal

This service is used when the ASPSP reports that the accessToken is expired. Through this request we can refresh the accessToken by sending the refreshToken associated with the expired accessToken.

6.2.1 Flow

Service to renew the access token, either the TPP access token or the HUB access token, when it has expired.

The next scenario shown in the image Figure 2: Access token renewal scenario occurs when the access token is expired, and it is necessary to perform the process to activate the access token again.

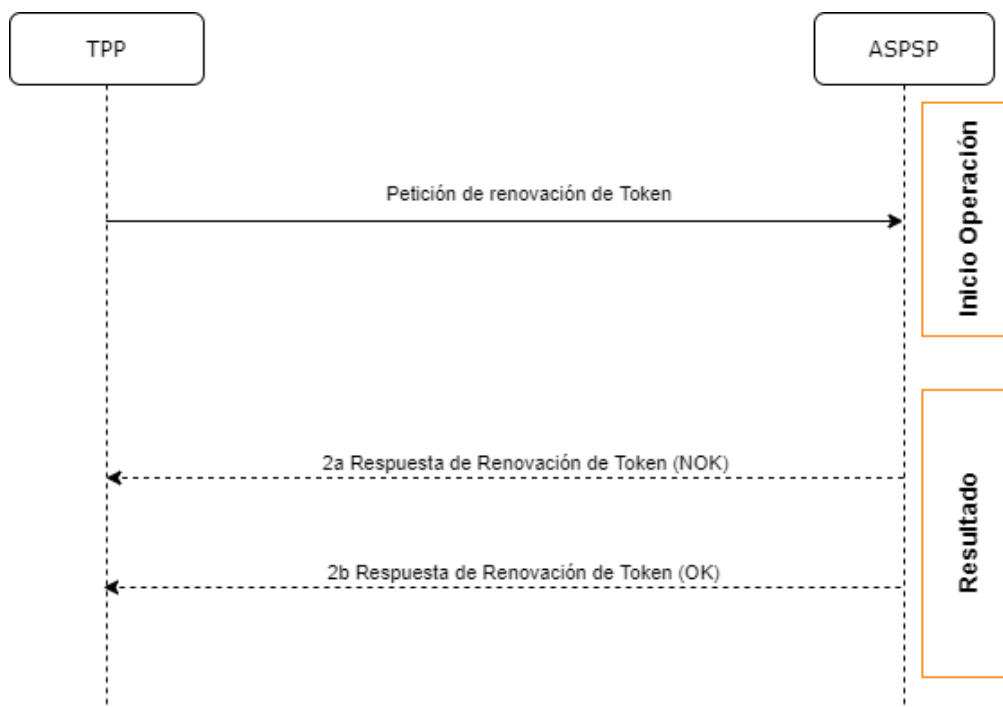


Figure 2: Access token renewal scenario

This process is described below:

1. Token Renewal Request (TPP ASPSP)

The TPP makes a POST request to the ASPSP's OAuth2 server to refresh the access token that will allow it to consume the displayed API services.

The ASPSP to renew the access token that will allow it to consume the entity's displayed API services.


```
POST /token HTTP/1.1
Host: aspsp.example.com
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW
Content-Type: application/x-www-form-urlencoded
```

grant_type=refresh_token&refresh_token=tGzv3JOkF0XG5Qx2TIKWIA

2. Token Renewal Response (ASPSP TPP)

The ASPSP evaluates the data provided by the TPP in the token renewal request and, if everything went correct, it will respond by renewing the token.

6.2.2 Request

Endpoint

POST {provider}/token

Path

Field	Description	Type	Mandat.	Format
provider	URL of the ASPSP where the service is published.	String	MA	Ex: aspsp.example.es
grant_type:	It must take the value of "refresh_token"	String	MA	Ex: grant_type=refresh_token
client_id	<p>"organizationIdentifier" provided in the eIDAS certificate formed as:</p> <ul style="list-style-type: none"> - PSD - 2 characters of the EQS country code according to ISO 3166 - Character "-" - 2-8 characters for NCA identifier 	String	MA	<p>^.{1,70}\$</p> <p>Ex: client_id=PSDES-RDS-4000</p>

	(AZ in uppercase) - Character "-" - PSP identifier			
refresh_token	Refresh token to obtain an unexpired accessToken.	String	MA	^.{1,64}\$ Ex: refresh_token=tGzv3JOkF0XG5Qx2TIKWI A

Header

No additional data is specified.

Body

No additional data is specified.

6.2.3 Response

Field	Description	Type	Mand at.	Format
access_token	Access token issued by the ASPSP and linked to the scope requested in the submission and confirmed by the PSU.	String	MA	^.{1,64}\$ Ex: "access_token": "83kdFZFEjr1zCsicMWBB"
token_type	Type of the issued token. It will take the value "Bearer".	String	MA	Ex: "token_type": "Bearer"
expires_in	Access token lifetime in seconds.	Integer	OP	Ex: "expires_in": 300
refresh_token	Refresh token. It can be used to obtain a new access token if it has expired.	String	OP	^.{1,64}\$ Ex: "refresh_token": "28JD3JOkF0NM5Qx2TICCC"

6.2.4 Examples

POST /token HTTP/1.1

Host: <https://hub.example.es>
Content-Type: application/x-www-form-urlencoded
grant_type = refresh_token & client_id = PSDES-RDS-4000 & refresh_token
= tGzv3JOkF0XG5Qx2TlKWIA

Example of OK response:

```
HTTP / 1.1 200 OK
Content-Type: application/json; charset=utf-8
Cache-Control: no-store
Pragma: no-cache
{
  "access_token": "83kdFZFEjrlzCsicMWBB",
  "token_type": "Bearer",
  "expires_in": 300,
  "refresh_token": "28JD3JOkF0NM5Qx2TlCCC"
}
```

6.3 PIS: Payment Initiation Service

6.3.1 Payment Initiation Flows

The payment initiation flow depends on the SCA approach implemented by the ASPSP.

Note: The flows do not always cover all the variations or complexities of the implementation and are sample flows.

6.3.1.1 SCA flow by redirection: implicit start of authorization process

The image below depicts Figure 3: Start of payment with OAuth2 as pre-step and SCA flow by redirection the sequence of requests/responses and redirections that are necessary in the flow where strong authentication (SCA) by redirection is needed (SCA over OAuth2 is not applied).

The characteristics of this flow are:

- TPP-Redirect-Preferred: true (SCA's TPP preference for redirection) or not reported (ASPSP decides for redirection)

PSD2 - APIs Implementation Guide v1.1 for TPPs

- TPP-Explicit-Authorization-Preferred: false - TPP preference to initiate the payment authorization process implicitly
- The PSU has only one SCA method

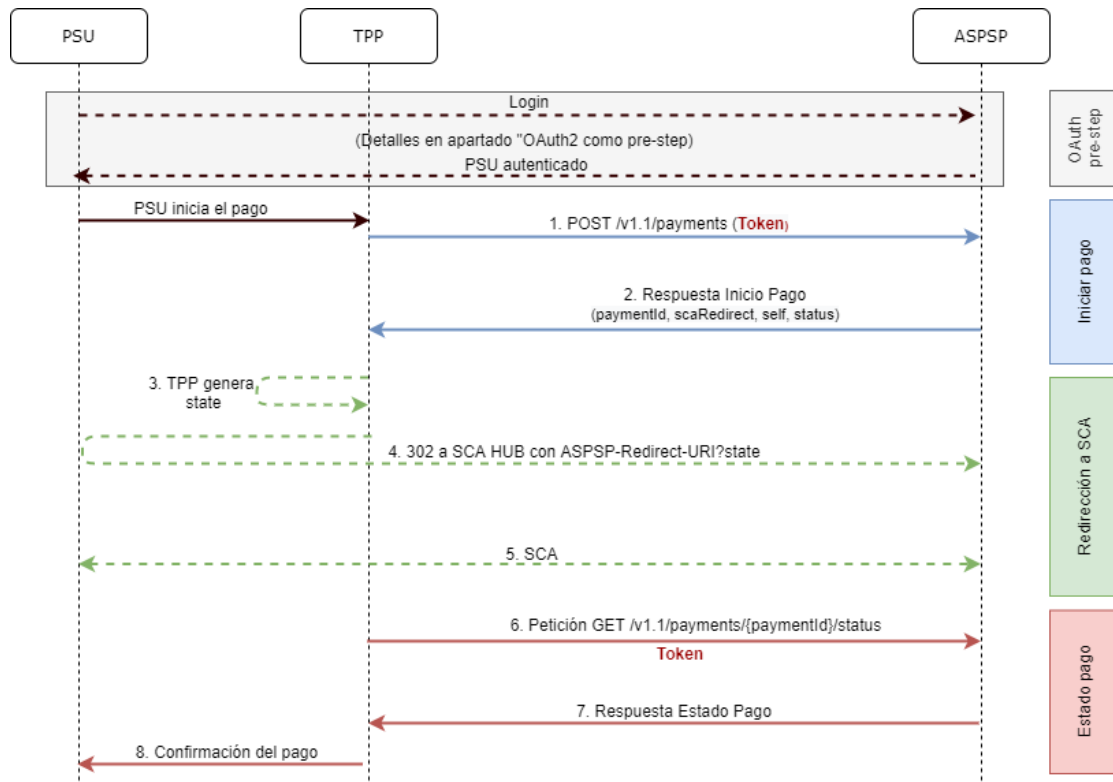


Figure 3: Start of payment with OAuth2 as pre-step and SCA flow by redirection and start of the implicit authorization process

OAuth2 (pre-step)

The main purpose of this flow is to authenticate the PSU to get access to the services displayed by its ASPSP through the use of an access token obtained after the application of this protocol.

In order to simplify, the detail of this flow has been omitted from the Figure 4 Figure 3: Start of payment with OAuth2 as pre-step and SCA flow by redirection and can be found in the section 6.16.1 OAuth2 as a pre-step.VERIFY

Note: this step is optional. Only applies if no valid access token is available.

PSU initiates payment

The PSU wants to pay through the TPP.

1. Start Payment Request (TPP → ASPSP)

The TPP sends a POST request to initiate payment with *token* to the ASPSP. Among the data reported by the TPP are:

- **TPP data:** identifier, name, roles, NCA, certificate ...
- **Payment data:** transfer type, ordering IBAN, beneficiary IBAN, amount, currency, concept ...
- **Data for risk scoring calculation:** IP, port, user-agent, language, location, HTTP headers ...
- **X-Request-ID:** identifier of the operation assigned by the TPP.
- **Access token** from TPP to Hub
- **TPP-Redirect-Preferred:** true (SCA flow preference by redirection) or not reported (ASPSP decides SCA by redirection).
- **TPP-Redirect-URI:** Return URI of the TPP after redirection to SCA.
- **TPP-Explicit-Authorization-Preferred:** false - TPP preference to initiate authorization implicitly
- **Other data**

2. Start Payment Response (ASPSP → TPP)

The ASPSP responds to the TPP indicating that strong authentication (SCA) is required, returning:

- **transactionStatus:** ISO 20022 state of the received payment start.
- **paymentId:** identifier of the generated resource that refers to the current payment initiation operation.
- **_links**
 - **scaRedirect:** links to the Hub endpoint where after receiving the redirect from the TPP it redirects back to the scaRedirect of the ASPSP. This URL can add security parameters to allow session maintenance during redirection.

`https://hub.example.com/auth`
 - **self:** link to the payment resource generated by the Hub for the payment initiation request received from the TPP.
 - **state:** link of the Hub to which the TPP can make a request to check the state of the payment.
- **Other data**

3. TPP generates state

The TPP, after receiving the response to initiate payment, generates a value for *state* (XSRF token) that it must link to the PSU browser session.

4. Redirect to scaRedirect Hub (TPP → ASPSP)

The TPP redirects the PSU to the authentication endpoint by adding to it the field *state* as a query-param.

```
HTTP/1.1 302 Found
Location: https://hub.example.com/auth?state=qwerty
```

5. SCA between PSU ↔ ASPSP

During this redirection process, the ASPSP will be able to:

- Show commissions to the PSU if required
- Show ASPSP-PSU interface for SCA

8. Payment State Request (TPP → ASPSP)

The TPP will send a request for payment state with *token* to know the state of the payment.

9. Payment State Response (ASPSP → TPP)

The ASPSP updates the state of the operation and responds to the TPP.

10. Payment confirmation

The TPP confirms the states of the payment to the PSU.

6.3.1.2 SCA flow by redirection: explicit start of authorization process.

Below in Figure 5: Initiation of payment with OAuth2 as pre-step and SCA flow by redirection and initiation of explicit authorisation process with/without selection of SCA method the sequence of requests/responses and redirections that are necessary in the flow where strong authentication (SCA) by redirection is needed (SCA over OAuth2 is not applied).

The characteristics of this flow are:

- TPP-Redirect-Preferred: true (SCA's TPP preference for redirection) or not reported (ASPSP decides for redirection)

PSD2 - APIs Implementation Guide v1.1 for TPPs

- TPP-Explicit-Authorization-Preferred: true - TPP's preference to start the payment authorization process explicitly or, if you select implicit and the PSU has more than one SCA method, the ASPSP switches to the explicit authorization process.

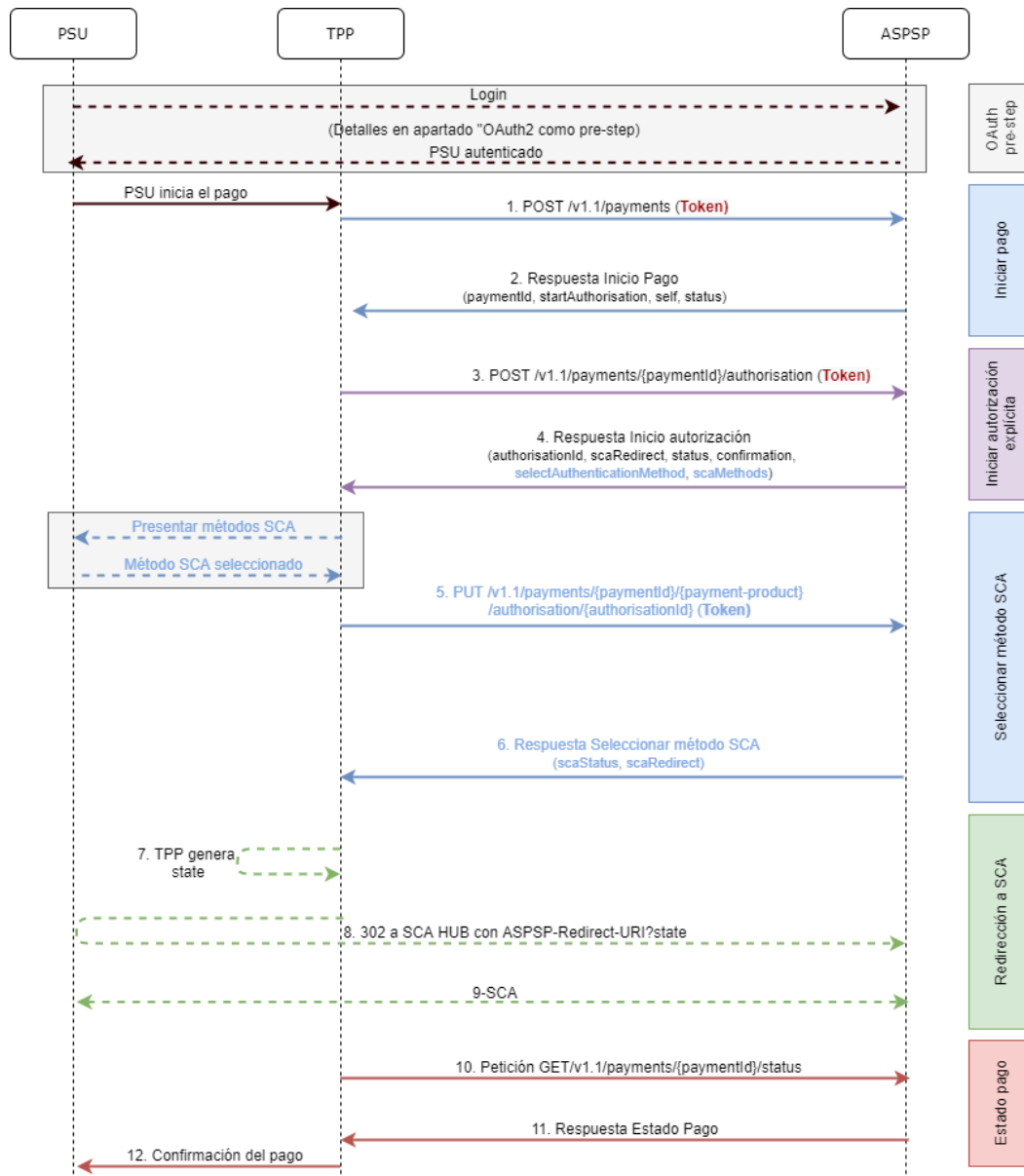


Figure 4: Start of payment with OAuth2 as pre-step and SCA flow by redirection and start of the explicit authorization process with / without selection of the SCA method

OAuth2 (pre-step)

The main purpose of this flow is to authenticate the PSU to get access to the services displayed by its ASPSP through the use of an access token obtained after the application of this protocol.

In order to simplify, the detail of this flow has been omitted from Figure 4: Payment initiation with OAuth2 as pre-step and SCA flow by redirection and can be found in the section 6.16.1 OAuth2 as a pre-step.VERIFY

Note: this step is optional. Only applies if no valid access token is available.

PSU initiates payment

The PSU wants to pay through the TPP.

1. Start Payment Request (TPP → ASPSP)

The TPP sends a POST request to initiate payment with *token* to the ASPSP. Among the data reported by the TPP are:

- **TPP data:** identifier, name, roles, NCA, certificate ...
- **Payment data:** transfer type, ordering IBAN, beneficiary IBAN, amount, currency, concept ...
- **Data for risk scoring calculation:** IP, port, user-agent, language, location, HTTP headers ...
- **X-Request-ID:** identifier of the operation assigned by the TPP.
- Hub access **token**
- **TPP-Redirect-Preferred:** true - SCA flow preference by redirect
- **TPP-Redirect-URI:** Return URI of the TPP after redirection to SCA.
- **TPP-Explicit-Authorization-Preferred:** true - TPP preference to initiate authorization explicitly (current flow)

- **Other data**

2. Initiate Payment Response (ASPSP → TPP)

The Hub, after receiving the response from the ASPSP, responds to the TPP by returning:

- **transactionStatus:** ISO 20022 state with the state of the transaction
- **paymentId:** resource identifier generated by the Hub referring to the current payment initiation transaction.

- **_links**
 - **self:** link to the resource that refers to the payment in the Hub
 - **state:** link of the Hub to which the TPP can make a request to check the state of the payment.

- **startAuthorisation:** link of the Hub to which the TPP can make a POST request to initiate the authorization of the payment explicitly.
- **Other data**

3. Initiate Authorization Request (TPP → ASPSP)

The TPP sends a POST request to initiate explicit authorization to initiate *token* payment to the ASPSP. Among the data reported by the TPP are:

- **TPP data:** identifier, name, roles, NCA, certificate ...
- **paymentId:** identifier of the payment to be authorized
- **X-Request-ID:** request identifier assigned by the TPP.
- **Access token** from the TPP

4. Initiate Authorization Response (ASPSP → TPP)

The ASPSP responds to the TPP indicating:

Response 1 - There is only one SCA method available, redirect to SCA is returned:

- **scaStatus:** state in which the SCA is.
- **authorizationId:** identifier of the authorization sub-resource created by the Hub
- **_links**
 - **scaRedirect:** link to the Hub's authentication server to start SCA through a redirect (SCA does not apply over OAuth2). This URL can add security parameters to allow session maintenance during redirection.

Ex: <https://hub.example.com/auth>

- **scaStatus:** link of the Hub to which the TPP can make a request to consult the state of the SCA of the corresponding authorization sub-resource.

Ex: <https://hub.example.com/v1.1/payments/{payment-product}/{paymentId}/authorisations/{authorisationId}>

- **Other data**

Response 2 - More than one SCA method available, selection by PSU necessary:

- **scaStatus:** state in which the SCA is.

- **authorizationId**: identifier of the authorization sub-resource created by the Hub
- **scaMethods**- Authentication objects that the PSU has available.
- **_links**
 - **selectAuthenticationMethod**: link of the Hub to which the TPP will be able to refer the SCA method selected by the PSU.

Ex: <https://hub.example.com/v1.1/payments/{payment-product}/{paymentId}/authorisations/{authorisationId}>
 - **scaStatus**: link of the Hub to which the TPP can make a request to consult the state of the SCA of the corresponding authorization sub-resource.

Ex: <https://hub.example.com/v1.1/payments/{payment-product}/{paymentId}/authorisations/{authorisationId}>
- **Other data**

Present SCA Methods (TPP → PSU) and select method (PSU → TPP)

The TPP, in case of receiving response 2 from the Hub (more than one SCA method), shows the PSU the SCA methods it has available to be selected.

The PSU selects one of the methods available to it.

5. Request Update PSU data (SCA Methods) (TPP → ASPSP)

The TPP sends a PUT request to update the SCA method selected by the PSU with *token* to the Hub. Among the data reported by the TPP are:

- **TPP data**: identifier, name, roles, NCA, certificate ...
- **authorizationId**: identifier of the authorization sub-resource created by the Hub
- **X-Request-ID**: request identifier assigned by the TPP.
- **Access token** from TPP to Hub
- **methodId**: identifier of the SCA method selected by the PSU

6. Response update PSU data (ASPSP → TPP)

The ASPSP responds to the TPP indicating:

- **scaStatus**: state in which the SCA is.

- **_links**

- **scaRedirect**: link to the Hub's authentication server to start SCA through a redirect (SCA does not apply over OAuth2). This URL can add security parameters to allow session maintenance during redirection.

Ex: <https://hub.example.com/auth>

- **scaStatus**: link of the Hub to which the TPP can make a request to consult the state of the SCA of the corresponding authorization sub-resource.

Ex: <https://hub.example.com/v1.1/payments/{payment-product}/{paymentId}/authorisations/{authorisationId}>

- **Other data**

7. TPP generates state

The TPP, after receiving the response, generates a value for *state* (XSRF token) to be linked to the PSU browser session.

8. Redirection to scaRedirect (TPP → ASPSP)

The TPP, after receiving the response to initiate authorization (or to update the SCA method), redirects the PSU to the authentication endpoint of the Hub and appends the *state* to it as query-param

HTTP/1.1 302 Found
Location: <https://hub.example.com/auth?state=qwerty>

SCA entre PSU ↔ ASPSP

During this redirection process, ASPSP will be able to show ASPSP-PSU interface for SCA

9. Payment State Request (TPP → ASPSP)

The TPP will send a tokenised payment state request to the ASPSP for payment state.

10. Payment State Response (ASPSP → TPP)

The ASPSP updates the state of the operation and responds to the TPP.

6.3.2 Payment start

Message sent by the TPP to the ASPSP through the Hub to create a payment start.

6.3.2.1 Request

Endpoint

POST {provider}/{aspsp}/v1.1/payments/{payment-product}

Path

Field	Description	Type	Mandat.	Format
provider	URL of the HUB where the service is released.	String	MA	Ex: www.hub.com
aspsp	Name of the ASPSP to which the request is to be made.	String	MA	Ex: aspsp-name
payment-product	Paid product to use. List of supported products: <ul style="list-style-type: none">• sepa-credit-transfers• instant-sepa-credit-transfers• target-2-payments• cross-border-credit-transfers	String	MA	Ex: {provider}/{aspsp}/v1.1/payments/sepa-credit-transfers/

Query parameters:

No additional parameters are specified for this request.

Header

Field	Description	Type	Mandat.	Format
-------	-------------	------	---------	--------

PSD2 - APIs Implementation Guide v1.1 for TPPs

Content-Type	Value: application / json	String	MA	Content-Type: application/json
X-Request-ID	Unique identifier of the operation assigned by the TPP.	String	MA	UUID <code>^[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}\$</code> Ex: X-Request-ID: 1b3ab8e8-0fd5-43d2-946e-d75958b172e7
Authorization	Bearer Token. Obtained in a previous authentication on OAuth2.	String	MA	Ex: Authorization: Bearer 2YotnFZFEjr1zCsicMWpAA
Consent-ID	This field will be ignored by the ASPSP. The session support is specified by the access token.	String	OP	<code>^.{1,36}\$</code> Ex: Consent-ID: 7890-asdf-4321
PSU-ID	Identifier that the PSU uses to identify itself in its ASPSP. It can be reported even if an OAuth token is being used and, in such a case, the ASPSP could check if the PSU-ID and the token match.	String	OP	Ex: PSU-ID: 12345678W
PSU-ID-Type	Type of the PSU-ID. Necessary in scenarios where the PSU has several PSU-IDs as access possibilities.	String	OP	Ex: PSU-ID-Type: NIF
PSU-Corporate-ID	Identifier of "company" in Online Channels.	String	OP	Ex: PSU-Corporate-ID: user@corporate.com

PSD2 - APIs Implementation Guide v1.1 for TPPs

PSU-Corporate-ID-Type	Type of the PSU-Corporate-ID required by the ASPSP to identify its content.	String	OP	Ex: PSU-Corporate-ID-Type: email
PSU-IP-Address	IP address of the HTTP request between the PSU and the TPP. If not available, the TPP should use the IP address used by the TPP when it sends this request.	String	MA	^[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\$ Ex: PSU-IP-Address: 192.168.16.5
TPP-Redirect-Preferred	If "true", the TPP has communicated to the HUB that it prefers SCA by redirection. If "false", the TPP has informed the HUB that it prefers not to be redirected to SCA and the procedure will be by decoupled flow. If the parameter is not used, the ASPSP will choose the SCA flow to apply depending on the SCA method chosen by the TPP / PSU. EMBEDDED NOT SUPPORTED IN THIS VERSION	Boolean	OP	Ex: TPP-Redirect-Preferred: true
TPP-Redirect-URI	URI of the TPP where the transaction flow must be redirected after any of the SCA phases. It is recommended to always use this header field. In the future, this field could change to mandatory.	String	COND	^.{1,250}\$ Ex: TPP-Redirect-URI: "https://tpp.example.es/cb"

TPP-Nok-Redirect-URI	If this URI is contained, the TPP is requesting to redirect the transaction flow to this address instead of the TPP-Redirect-URI in case of a negative result of the SCA method by redirection.	String	OP	^.{1,250}\$ Ex: TPP-Nok-Redirect-URI:"https://tpp.example.es/cb/nok"
TPP-Explicit-Authorisation-Preferred	If equal to true, the TPP chooses to initiate the authorisation process separately, e.g. due to the need for authorisation of a set of operations simultaneously. If false or the parameter is not used, there is no TPP preference. The TPP takes a direct authorisation of the transaction in the next step. Note: ASPSP might not take it into account if it doesn't support it.	Boolean	OP	Ex: TPP-Explicit-Authorisation-Preferred: false
TPP-Brand-Logging-Information	This field could be used by the TPP to inform the ASPSP about the brand used by the TPP for the PSU. This information can be used to improve communication between the ASPSP and the PSU or the ASPSP and the TPP.	String	OP	^.{1,70}\$ Ex: TPP-Brand-Logging-Information: TPP Brand

Body

The content of the Body is defined in 8.14 SinglePayment and the following parameters included in the table below.

Fields marked as mandatory (MA) and optional (OP) are supported by the ASPSP with that type of condition.

Fields marked as COND are dependent on each ASPSP.

Campo	SCT	SCT INST	Target 2	Cross Border CT
debtorName	COND	COND	COND	COND
debtorAccount	MA	MA	MA	MA
instructedAmount	MA	MA	MA	MA
creditorAccount	MA	MA	MA	MA
creditorAgent	OP	OP	OP	MA
creditorName	MA	MA	MA	MA
creditorAddress	OP	OP	OP	OP
chargeBearer	COND	COND	COND	COND
remittanceInformationUnstructured	OP	OP	OP	OP
remittanceInformationStructured	COND	COND	COND	COND
requestedExecutionDate	n.a.	n.a.	n.a.	n.a.

6.3.2.2 Response

Header

Field	Description	Type	Mandat.	Format
Location	Contains the link to the generated resource.	String	MA	$\wedge.\{1,512\}\$$ Ex: Location: /v1.1/payments/{payment-product}/{payment-id}
X-Request-ID	Unique identifier of the operation assigned by the TPP.	String	MA	UUID $\wedge[0-9a-fA-F]\{8\}-[0-9a-fA-F]\{4\}-[0-9a-fA-F]\{4\}-[0-9a-fA-F]\{4\}-[0-9a-fA-F]\{12\}\$$ Ex:

				X-Request-ID: 1b3ab8e8-0fd5- 43d2-946e- d75958b172e7
ASPSP-SCA-Approach	Value returned if the SCA method has been set. Possible values: <ul style="list-style-type: none">• REDIRECT The OAuth based SCA will be taken as REDIRECT.	String	COND	Ex: ASPSP-SCA-Approach: REDIRECT

Body

Field	Description	Type	Mandat.	Format
transactionStatus	Transaction state. Values defined in annexes in 9.4 Transaction states	String	MA	ISO 20022 Ex: "transactionStatus": "RCVD"
paymentId	Resource identifier that refers to the initiation of payment.	String	MA	^.{1,36}\$ Ex: "paymentId": "1b3ab8e8-0fd5- 43d2-946e- d75958b172e7"
_links	List of hyperlinks to be recognized by the TPP. Supported types in this response: <ul style="list-style-type: none">• scaRedirect: in case of SCA by redirection. Link where the PSU browser must be redirected by the TPP.• startAuthorisation: in case an explicit start of transaction authorization is required (no SCA method selection)	Links	MA	Ex: "_links": {...}

	<ul style="list-style-type: none"> • self: link to the resource created by this request. • state: link to retrieve the state of the transaction. • scaStatus: link to consult the SCA state corresponding to the authorisation sub-resource. This link is only contained if an authorization sub-resource has been created. 			
psuMessage	Text to show to the PSU.	String	OP	$\wedge.\{1,500\}$ \$ Ex: "psuMessage": "Información para PSU"
tppMessages	Message to the TPP	List<Tpp Message >	OP	Ex: "tppMessages": [...]

6.3.2.3 Examples

Example request for SCA by redirection

POST <https://www.hub.com/aspsp-name/v1.1/payments/sepa-credit-transfers>

Content-Encoding: gzip

Content-Type: application/json

X-Request-ID: 10391c7e-ad88-49ec-a2ad-00aacb1f6541

Authorization: Bearer 2YotnFZFjrlzCsicMWpAA

PSU-IP-Address: 192.168.8.16

PSU-IP-Port: 443

PSU-Accept: application/json

PSU-Accept-Charset: utf-8

PSU-Accept-Encoding: gzip

PSD2 - APIs Implementation Guide v1.1 for TPPs

```
PSU-Accept-Language: es-ES
PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0
PSU-Http-Method: POST
PSU-Device-ID: f8b3feda-6fe3-11e8-adc0-fa7ae01bbebc
PSU-GEO-Location: GEO:12.526347;54.649862
TPP-Redirect-Preferred: true
TPP-Redirect-URI: https://www.tpp.com/cb
TPP-Nok-Redirect-URI: https://www.tpp.com/cb/nok
Date: Sun, 26 Sep 2017 15:02:37 GMT
{
  "instructedAmount": {
    "currency": "EUR",
    "amount": "153.50"
  },
  "debtorAccount": {
    "iban": "ES1111111111111111111111"
  },
  "creditorAccount": {
    "iban": "ES2222222222222222222222"
  },
  "creditorName": "Nombre123",
  "remittanceInformationUnstructured": "Información adicional"
}
```

Example response in case of SCA by redirection with an implicitly created authorization sub-resource

```
HTTP/1.1 201 Created
X-Request-ID: 10391c7e-ad88-49ec-a2ad-00aacb1f6541
ASPSP-SCA-Approach: REDIRECT
Date: Sun, 26 Sep 2017 15:02:43 GMT
Location: /v1.1/payments/sepa-credit-transfers/123-qwe-456
Content-Type: application/json
{
  "transactionStatus": "RCVD",
  "paymentId": "123-qwe-456",
```

```
"_links": {
  "scaRedirect": {
    "href": "https://hub.example.es/authorize "
  },
  "self": {
    "href": "/v1.1/payments/sepa-credit-transfers/123-qwe-456",
    "state": {
      "href": "/v1.1/payments/sepa-credit-transfers/123-qwe-456/state"
    },
    "scaStatus": {
      "href": "/v1.1/payments/sepa-credit-transfers/123-qwe-456/authorisations/123auth456"
    }
  }
}
```

6.3.3 Future payment start

Message sent by the TPP to the ASPSP to create a future payment start.

This functionality is similar to the Payment Initiation. The only difference that exists is in the messaging of the Start of payment request that supports an optional extra parameter "requestedExecutionDate" to indicate the future date on which the payment would be executed.

In this type of payment, after the execution of SCA, the payment is not executed, but the ASPSP leaves it scheduled to execute on the specified date.

6.3.3.1 Request

Endpoint

POST {provider}/{aspsp}/v1.1/payments/{payment-product}

Path

Field	Description	Type	Mandat.	Format
-------	-------------	------	---------	--------

PSD2 - APIs Implementation Guide v1.1 for TPPs

provider	URL of the HUB where the service is released.	String	MA	Ex: hub.example.es
aspsp	Name of the ASPSP to which the request is to be made.	String	MA	Ex: aspsp-name
payment-product	Paid product to use. List of supported products: <ul style="list-style-type: none">sepa-credit-transfers	String	MA	Ex: {provider}/{aspsp}/v1.1/payments/sepa-credit-transfers/

Query parameters:

No additional parameters are specified for this request.

Header

The same as those defined in the section 6.3.2.1

Body

The content of the Body is the one defined in 8.14 SinglePayment and the following parameter must also be reported:

Field	Description	Type	Mand at.	Format
requestedExecutionDate	The payment will be executed on the informed date. Note: this field must be reported.	String	OP	ISODate Ex: "requestedExecutionDate": "2019-01-12"

6.3.3.2 Response

HTTP Code

201 if the resource has been created

Header

The same as those defined in the section 6.3.2.2

Body

Field	Description	Type	Mandat.	Format
transactionStatus	Transaction state. Values defined in annexes in 9.4 Transaction states	String	MA	ISO 20022 Ex: "transactionStatus": "RCVD"
paymentId	Resource identifier that refers to the initiation of payment.	String	MA	^.{1,36}\$ Ex: "paymentId": "1b3ab8e8-0fd5-43d2-946e-d75958b172e7"
_links	<p>List of hyperlinks to be recognized by the TPP. Supported types in this response:</p> <ul style="list-style-type: none"> • scaRedirect: in case of SCA by redirection. Link where the PSU browser must be redirected by the TPP. • startAuthorisation: in case an explicit start of transaction authorization is required (no SCA method selection) • self: link to the resource created by this request. • state: link to retrieve the state of the transaction. • scaStatus: link to consult the SCA state corresponding to the authorisation sub-resource. This link is only contained if an 	Links	MA	Ex: "_links": {...}

PSD2 - APIs Implementation Guide v1.1 for TPPs

	authorization sub-resource has been created.			
psuMessage	Text to show to the PSU.	String	OP	^.{1,500} \$ Ex: "psuMessage": "Información para PSU"
tppMessages	Message to the TPP	List<Tpp Message>	OP	Ex: "tppMessages": [...]

6.3.3.3 Examples

Example request for SCA by redirection

POST <https://hub.example.es/aspsp-name/v1.1/payments/sepa-credit-transfers>

Content-Encoding: gzip

Content-Type: application/json

X-Request-ID: 10391c7e-ad88-49ec-a2ad-00aacblf6541

Authorization: Bearer 2YotnFZFEjrlzCsicMWpAA

PSU-IP-Address: 192.168.8.16

PSU-IP-Port: 443

PSU-Accept: application/json

PSU-Accept-Charset: utf-8

PSU-Accept-Encoding: gzip

PSU-Accept-Language: es-ES

PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0) Gecko/20100101 Firefox/54.0

PSU-Http-Method: POST

PSU-Device-ID: f8b3feda-6fe3-11e8-adc0-fa7ae01bbebc

PSU-GEO-Location: GEO:12.526347;54.649862

TPP-Redirect-Preferred: true

TPP-Redirect-URI: https://tpp.example.es/cb

TPP-Nok-Redirect-URI: https://tpp.example.es/cb/nok

Date: Sun, 26 Sep 2017 15:02:37 GMT

```
{
  "instructedAmount": {
```

```
        "currency": "EUR",
        "amount": "153.50"
    },
    "debtorAccount": {
        "iban": "ES1111111111111111111111"
    },
    "creditorAccount": {
        "iban": "ES2222222222222222222222"
    },
    "creditorName": "Nombre123",
    "remittanceInformationUnstructured": "Información adicional",
    "requestedExecutionDate": "2019-01-12"
}
```

6.3.4 Initiation of standing orders for recurring/periodic payments

Message sent by the TPP to the ASPSP to create a recurring / periodic payment start.

The recurring payment initiation functionality is covered by the Berlin Group specification as the initiation of a specific standing order.

A TPP can send a recurring payment start where the start date, frequency and, conditionally, end date are provided.

Once authorized by the PSU, the payment will be executed by the ASPSP, if possible, following the "standing order" as it was sent by the TPP. No further action is required from the TPP.

In this context, this payment is considered a periodic payment to differentiate the payment from other types of recurring payments where third parties are initiating the same amount of money.

Note: for standing orders of payment initiations, the ASPSP will always ask for SCA with Dynamic linking. No exemptions are allowed.

Reglas campo dayOfExecution

- **Monthly payments or higher:** possible values range from 01 to 31. Using 31 as the last day of the month

6.3.4.1 Request

Endpoint

POST {provider}/{aspsp}/v1.1/periodic-payments/{payment-product}

Path

Field	Description	Type	Mandat.	Format
provider	URL of the HUB where the service is released.	String	MA	Ex: hub.example.es
aspsp	Name of the ASPSP to which the request is to be made.	String	MA	Ex: aspsp-name
payment-product	Paid product to use. List of supported products: <ul style="list-style-type: none"> sepa-credit-transfers 	String	MA	Ex: {provider}/{aspsp-name)/v1.1/periodic-payments/sepa-credit-transfers/

Query parameters:

No additional parameters are specified for this request.

Header

The same as those defined in the section 6.3.2.1

Body

The content of the Body is defined in 8.14 SinglePayment plus those defined below:

Field	Description	Type	Mandat.	Format
startDate	The first applicable day of execution from this date is the first payment	String	MA	ISODate xEx: "startDate":"2018-12-20"
execution Rule	Supported values: <ul style="list-style-type: none"> following preceding 	String	OP	Ex: "executionRule":"following"

	<p>Defines the behavior when recurring payment dates fall on weekends or holidays. Payment is then executed on the preceding or following working day.</p> <p>The ASPSP may reject the request due to the communicated value if the Online Banking rules do not support this execution rule.</p>			
endDate	<p>The last applicable day of execution.</p> <p>If not given, it is an endless standing order.</p>	String	OP	<p>ISODate</p> <p>Ex: "endDate":"2019-01-20"</p>
frequency	<p>The frequency of the recurring payment resulting from this standing order.</p> <p>Allowed values:</p> <ul style="list-style-type: none"> • Monthly • Quarterly • Semi Annual 	String	MA	<p>EventFrequency7Code de ISO 20022</p> <p>Ex: "frequency": "Monthly"</p>
dayOfExecution	<p>"31" is last.</p> <p>Following the regular expression \d{1,2}</p> <p>The date refers to the ASPSP time zone.</p> <p>Only if supported in ASPSP Online Banking.</p>	String	COND	<p>\d{1,2}</p> <p>Ex: "dayOfExecution": "01"</p>

6.3.4.2 Response

HTTP Code

201 if the resource has been created

Header

The same as those defined in the section 6.3.2.2

Body

The same as those defined in the section 6.3.2.2

6.3.4.3 Examples

Example request for SCA by redirection

POST <https://hub.example.es/{aspsp-name}/v1.1/periodic-payments/sepa-credit-transfers>

Content-Encoding: gzip

Content-Type: application/json

X-Request-ID: 10391c7e-ad88-49ec-a2ad-00aacb1f6541

Authorization: Bearer 2YotnFZFEjrlzCsicMWpAA

PSU-IP-Address: 192.168.8.16

TPP-Redirect-Preferred: true

TPP-Redirect-URI: https://tpp.example.es/cb

TPP-Nok-Redirect-URI: https://tpp.example.es/cb/nok

Date: Sun, 26 Sep 2017 15:02:37 GMT

```
{
  "instructedAmount": {
    "currency": "EUR",
    "amount": "153.50"
  },
  "creditorAccount": {
    "iban": "ES2222222222222222222222"
  },
  "creditorName": "Nombre123",
  "remittanceInformationUnstructured": "Información adicional"
  "startDate": "2018-03-01",
  "executionRule": "preceeding",
  "frequency": "Monthly",
  "dayOfExecution": "01"
```

}

6.3.5 Get payment state

This message is sent by the TPP to the HUB to request information on the state of the payment initiation requested by the TPP.

6.3.5.1 Request

Endpoint

GET {provider}/{aspsp}/v1.1/{payment-service}/{payment-product}/{paymentId}/state

Path

Field	Description	Type	Mandant.	Format
provider	URL of the HUB where the service is released.	String	MA	Ex: www.hub.com
aspsp	Name of the ASPSP to which the request is to be made.	String	MA	Ex: aspsp-name
payment-service	Possible values are: <ul style="list-style-type: none"> payments periodic-payments 	String	MA	Ex: {provider} / {aspsp} /v1.1/payments
payment-product	Paid product to use. List of supported products: <ul style="list-style-type: none"> sepa-credit-transfers instant-sepa-credit-transfers target-2-payments cross-border-credit-transfers 	String	MA	Ex: {provider}/{aspsp}/v1.1/payments/sepa-credit-transfers/
paymentId	Resource identifier that refers to the initiation of payment.	String	MA	^. {1,36}\$ Ex: 1234-qwer-5678

	Previously sent in response to a payment initiation message from the TPP to the HUB.			
--	--	--	--	--

Query parameters:

No additional fields are specified.

Header

Field	Description	Type	Mandate	Format
X-Request-ID	Unique identifier of the request assigned by the TPP.	String	MA	UUID ^[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}\$ Ex: X-Request-ID: 1b3ab8e8-0fd5-43d2-946e-d75958b172e7
Authorization	Bearer Token. Obtained in a previous authentication on OAuth2.	String	MA	Ex: Authorization: Bearer 2YotnFZFEjr1zCsicMWpAA
Accept	Supported format of the response. Supported values: <ul style="list-style-type: none"> application/json 	String	OP	^.{1,50}\$ Ex: Accept: application/json

Body

No additional data is specified.

6.3.5.2 Response

Header

Field	Description	Type	Mandate	Format
X-Request-ID	Unique identifier of the request assigned by the TPP.	String	MA	UUID $^{\wedge}[0-9a-fA-F]\{8\}-[0-9a-fA-F]\{4\}-[0-9a-fA-F]\{4\}-[0-9a-fA-F]\{4\}-[0-9a-fA-F]\{12\}\$$ Ex: X-Request-ID: 1b3ab8e8-0fd5-43d2-946e-d75958b172e7

Body

Field	Description	Type	Mandate	Format
transactionStatus	State of the payment transaction. Defined values in 9.4 Transaction states	String	MA	ISO20022 Ex: "transactionStatus": "ACCP"
psuMessage	Text to show to the PSU.	String	OP	$^{\wedge}.\{1,500\}\$$ Ex: "psuMessage": "Información para PSU"
tppMessages	Message to the TPP	List<Tp pMessage>	OP	Ex: "tppMessages": [...]

6.3.5.3 Examples

Example of request

GET <https://www.hub.com/aspsp-name/v1.1/payments/sepa-credit-transfer/123asdf456/state>

Accept: application/json

X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc

Authorization: Bearer 2YotnFZFEjrlzCsicMWpAA

PSU-IP-Address: 192.168.8.16

PSU-IP-Port: 443

PSU-Accept: application/json

PSD2 - APIs Implementation Guide v1.1 for TPPs

```
PSU-Accept-Charset: utf-8
PSU-Accept-Encoding: gzip
PSU-Accept-Language: es-ES
PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0
PSU-Http-Method: GET
PSU-Device-ID: f8b3feda-6fe3-11e8-adc0-fa7ae01bbebc
PSU-GEO-Location: GEO:12.526347;54.649862
Date: Sun, 26 Sep 2017 15:02:48 GMT
```

Example response

```
HTTP/1.1 200 Ok
X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc
Date: Sun, 26 Sep 2017 15:02:50 GMT
Content-Type: application/json
{
  "transactionStatus": "ACCP",
  "fundsAvailable": true
}
```

6.3.6 Retrieve payment initiation information

This message is sent by the TPP through the HUB to the ASPSP to obtain the information of a payment initiation.

6.3.6.1 Request

Endpoint

GET {provider}/{aspsp}/v1.1/{payment-service}/{payment-product}/{paymentId}

Path

Field	Description	Type	Mandat.	Format
provider	URL of the HUB where the service is released.	String	MA	Ex: www.hub.com

PSD2 - APIs Implementation Guide v1.1 for TPPs

aspsp	Name of the ASPSP to which the request is to be made.	String	MA	Ex: aspsp-name
payment-service	Possible values are: <ul style="list-style-type: none">• payments• periodic-payments	String	MA	Ex: {provider} / {aspsp} /v1.1/payments
payment-product	Paid product to use. List of supported products: <ul style="list-style-type: none">• sepa-credit-transfers• instant-sepa-credit-transfers• target-2-payments• cross-border-credit-transfers	String	MA	Ex: {provider}/{aspsp}/v1.1/payments/sepa-credit-transfers/
paymentId	Resource identifier that refers to the initiation of payment. Previously sent in response to a payment initiation message from the TPP to the HUB.	String	MA	^. {1,36}\$ Ex: 1234-qwer-5678

Query parameters:

No additional fields are specified.

Header

The same as those defined in the section 6.3.5.1

Body

No additional fields are specified.

6.3.6.2 Response

Header

The same as those defined in the section 6.3.5.2

Body

PSD2 - APIs Implementation Guide v1.1 for TPPs

The fields to be returned are those of the original payment initiation request:

- 6.3.2 Payment start
- 6.3.3 Future payment start
- 6.3.4 Initiation of standing orders for recurring/periodic payments

Plus the following

Note 1: the debtorName must be included even if it was not sent by the TPP. In this way the ASPSP will be able to return the name of the PSU to the PISP due to regulatory needs.

Note 2: according to item 40 of [EBA-OP2], the payment resource must contain the debtorAccount after the payment has been correctly initiated, even if it has not been sent by the TPP.

Field	Description	Type	Mandat.	Format
transactionStatus	Transaction state. Values defined in annexes. Short Code.	String	MA	ISO 20022 Ex: "transactionStatus": "ACCP"
debtorName	Name of the PSU. In case it is not provided by the TPP, the ASPSP may return it for regulatory needs.	String	OP	^.{1, 70}\$ Ex: "debtorName": "Paul Simpson"
psuMessage	Text sent to the TPP through the HUB to be displayed to the PSU.	String	OP	^.{1,500} \$ Ex: "psuMessage": "Información para PSU"
tppMessages	Message for the TPP sent through the HUB.	List<message>	OP	Ex: "tppMessage": [...]

6.3.6.3 Examples

Example of request

GET <https://www.hub.com/aspsp-name/v1.1/payments/sepa-credit-transfers/123-asdf-456>

Accept: application/json

X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc

PSD2 - APIs Implementation Guide v1.1 for TPPs

Authorization: Bearer 2YotnFZFEjrlzCsicMWpAA
PSU-IP-Address: 192.168.8.16
PSU-IP-Port: 443
PSU-Accept: application/json
PSU-Accept-Charset: utf-8
PSU-Accept-Encoding: gzip
PSU-Accept-Language: es-ES
PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0
PSU-Http-Method: GET
PSU-Device-ID: f8b3feda-6fe3-11e8-adc0-fa7ae01bbebc
PSU-GEO-Location: GEO:12.526347;54.649862
Date: Sun, 26 Sep 2017 15:02:48 GMT

Example response

HTTP/1.1 200 Ok
X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc
Date: Sun, 26 Sep 2017 15:02:50 GMT
Content-Type: application/json
{
 "instructedAmount": {
 "currency": "EUR",
 "amount": "153.50"
 },
 "debtorAccount": {
 "iban": "ES1111111111111111111111"
 },
 Ex: "debtorName": "Paul Simpson"
 "creditorAccount": {
 "iban": "ES2222222222222222222222"
 },
 "creditorName": "Nombre123",
 "remittanceInformationUnstructured": "Información adicional",
 "transactionStatus": "ACCP",
}

6.3.7 Cancel start of payment

This request is sent by the TPP to the ASPSP through the Hub and allows to initiate the cancellation of a payment. Depending on the payment service, the payment product, and the ASPSP implementation, this request may be sufficient to cancel the payment or an authorization may be required.

6.3.7.1 Request

Endpoint

DELETE {provider}/{aspsp}/v1.1/{payment-service}/{payment-product}/{paymentId}

Path

Field	Description	Type	Mand at.	Format
provider	URL of the ASPSP where the service is published.	String	MA	Ex: www.hub.com
aspsp	Name of the ASPSP to which the request is to be made.	String	MA	Ex: aspsp-name
payment-service	Possible values are: <ul style="list-style-type: none"> payments periodic-payments 	String	MA	Ex: {provider}/v1.1/payments
paymentId	Resource identifier that refers to the initiation of payment. Previously sent in response to a payment initiation message from the HUB to the ASPSP.	String	MA	^. {1,36}\$ Ex: 123-qwe-456

Query parameters:

No additional fields are specified.

Header

Field	Description	Type	Mandat.	Format
X-Request-ID	Unique identifier of the request assigned by the TPP.	String	MA	UUID $\wedge[0-9a-fA-F]\{8\}-[0-9a-fA-F]\{4\}-[0-9a-fA-F]\{4\}-[0-9a-fA-F]\{4\}-[0-9a-fA-F]\{12\}\$$ Ex: X-Request-ID: 1b3ab8e8-0fd5-43d2-946e-d75958b172e7
Authorization	Bearer Token. Obtained in a previous authentication on OAuth2.	String	MA	Ex: Authorization: Bearer 2YotnFZFEjr1zCsicM WpAA

Body

No additional data is specified.

6.3.7.2 Response

HTTP Code

204 if the DELETE command is sufficient to cancel the payment.

Header

The same as those defined in the section 6.3.5.2

Body

Field	Description	Type	Mandat.	Format
-------	-------------	------	---------	--------

transactionStatus	Transaction state. Values defined in annexes in Error! Reference source not found 9.4 Transaction states	String	MA	ISO 20022 Ex: "transactionStatus": "CANC"
psuMessage	Text sent to the TPP through the HUB to be displayed to the PSU.	String	OP	^.{1,500} \$ Ex: "psuMessage": "Información para PSU"
tppMessages	Message for the TPP sent through the HUB.	List<Tp pMessage>	OP	Ex: "tppMessages": [...]

6.3.7.3 Examples

Example of request

```
DELETE https://www.hub.com/aspsp-name/v1.1/payments/sepa-credit-transfers/123-qwe-456
X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc
Authorization: Bearer 2YotnFZFEjrlzCsicMWpAA
PSU-IP-Address: 192.168.8.16
PSU-IP-Port: 443
PSU-Accept: application/json
PSU-Accept-Charset: utf-8
PSU-Accept-Encoding: gzip
PSU-Accept-Language: es-ES
PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0
PSU-Http-Method: GET
PSU-Device-ID: f8b3feda-6fe3-11e8-adc0-fa7ae01bbebc
PSU-GEO-Location: GEO:12.526347;54.649862
TPP-Redirect-Preferred: true
TPP-Redirect-URI: https://www.tpp.com/cb
TPP-Nok-Redirect-URI: https://www.tpp.com/cb/nok
Ex: TPP-Explicit-Authorisation-Preferred: false
Content-Type: application/json
Date: Sun, 26 Sep 2017 15:02:48 GMT
```

Example response

HTTP / 1.1 204 No content

X-Request-ID: 0ee25bf4-6ff1-11e8-adc0-fa7ae01bbebc

Date: Sun, 26 Sep 2017 15:02:47 GMT

6.4 AIS: Service to establish consent of information about accounts

6.4.1 Characteristics of consent

6.4.1.1 Consent model

Sample	Description
Detailed consent (Detailed consent)	<p>Request consent on specified accounts</p> <p>Create a consent, which the ASPSP must store, requesting access to the specified accounts and with the requested access.</p> <p>If a consent already existed, said consent will expire and the new one will take effect when authorized by the USP.</p> <p>The accounts for which consent is requested to access "balances" and / or "transactions", it is assumed that they will also have the access type "accounts".</p>
Global consent (Global consent)	<p>Request consent on the list of available accounts</p> <p>This functionality only serves to request consent to the list of available accounts of the PSU. It does not give consent for "accounts", "balances" and / or "transactions".</p> <p>This request does not indicate the accounts to which access is sought. It is specified to be requested for "all available accounts" by indicating in the access the attribute "availableAccounts" or "availableAccountsWithBalance" with the value "allAccounts".</p> <p>It is a one-time consent to obtain the list of available accounts. It will not give the details of the accounts.</p> <p>Request consent to access all accounts for all PSD2 AIS services</p> <p>Request access for all available PSU accounts on all PSD2 AIS services.</p>

	<p>The accounts are not given by the TPP.</p> <p>This request does not indicate the accounts to which access is sought. It is specified to be requested for "all PSD2 accounts" by indicating in the access the attribute "allPsd2" with the value "allAccounts".</p> <p>The TPP, through the HUB, can retrieve such information managed between ASPSP and PSU by making a request to retrieve consent information.</p>
Consent offered by the bank (Bank offered consent)	<p>Request consent without indicating accounts</p> <p>Request consent for access to "accounts", "balances" and / or "transactions" without indicating the accounts. That is, the attributes "accounts", "balances" and "transactions" will be with a blank array.</p> <p>To select the accounts for which access is to be provided, access must be obtained bilaterally between ASPSP and PSU via the ASPSP interface in the OAuth redirect flow.</p> <p>The ASPSP in the redirection process will show the PSU its accounts to choose which ones it wants to consent to the TPP.</p> <p>The TPP, through the HUB, can retrieve such information managed between ASPSP and PSU by making a request to retrieve consent information.</p>

6.4.1.2 Recurrence in access

Recurring consent

If a prior consent with recurring access already exists (recurringIndicator = true) and a new consent request with recurring access is sent, as soon as the new consent is accepted by the PSU, the previous consent will expire and the valid consent will be the new consent requested.

A consent with recurring access may have one or more accounts with different types of access ("accounts", "balances", "transactions").

Note: giving access to "balances" and / or "transactions" automatically grants access to "accounts" to said accounts

Non-recurring consent

A consent request for a non-recurring access (one-time use and recurringIndicator=false) will be treated as a new consent (new consentId) without affecting previous existing consent.

6.4.1.3 Return of the account holder's name

This specification is based on one of the consent models described in NextGenPSD2 XS2A Framework v1.3.8. In particular, the following model is used for this specification:

- The ASPSP will release the name of the account holder, in this case, the name of the connected PSU, without adopting the extension of the consent model defined in the standard.

6.4.1.4 List of standing orders

Obtain the list of standing orders for a specific account. Information is returned as transactions using the "bookingStatus" entry state with the value "information".

6.4.1.5 List of trusted payees

This specification is based on one of the consent models described in NextGenPSD2 XS2A Framework - Extended IG Trusted Beneficiaries v1. In particular, the following model is used for this specification:

- The ASPSP will release the list of trusted payees without adopting the extension of the consent model defined in the standard.

Note: the global consent model covers this type of access.

6.4.1.6 Consent state information

The state of the consent resource changes during the process of establishing consent. The attribute defined for the consent state is defined as "consentStatus".

The only states supported in the initial phase for consentStatus are "received", "rejected" and "valid".

After successful authorization by a PSU, the consent resource could change its state during its life cycle. The following codes are supported during the consent lifecycle phase:

- "expired": consent has expired (for example, after 90 days)
- "revokedByPsu": consent has been revoked by the PSU
- "terminatedByTpp": the TPP has terminated consent

The TPP can retrieve this state on the GET request to retrieve consent state.

Note: the "expired" state also applies to single-use consent, once they have been used or have expired.

Note: the "terminatedByTpp" state also applies when a recurring consent has been terminated by the TPP by establishing a new recurring consent.

Additionally, the TPP can retrieve the SCA state for consent establishment with the corresponding SCA state GET request.

6.4.2 Account information consent flows

6.4.2.1 SCA flow by redirection: implicit start of authorization process

The image below depicts Figure 5: SCA flow by redirection: implicit start of the authorisation process the sequence of requests/responses and redirections that are necessary in the flow where strong authentication (SCA) by redirection is needed (SCA over OAuth2 is not applied).

The characteristics of this flow are:

- TPP-Redirect-Preferred: true - SCA TPP preference for redirection
- TPP-Explicit-Authorisation-Preferred: false - TPP preference to initiate the authorisation process associated with consent implicitly
- The PSU has only one SCA method

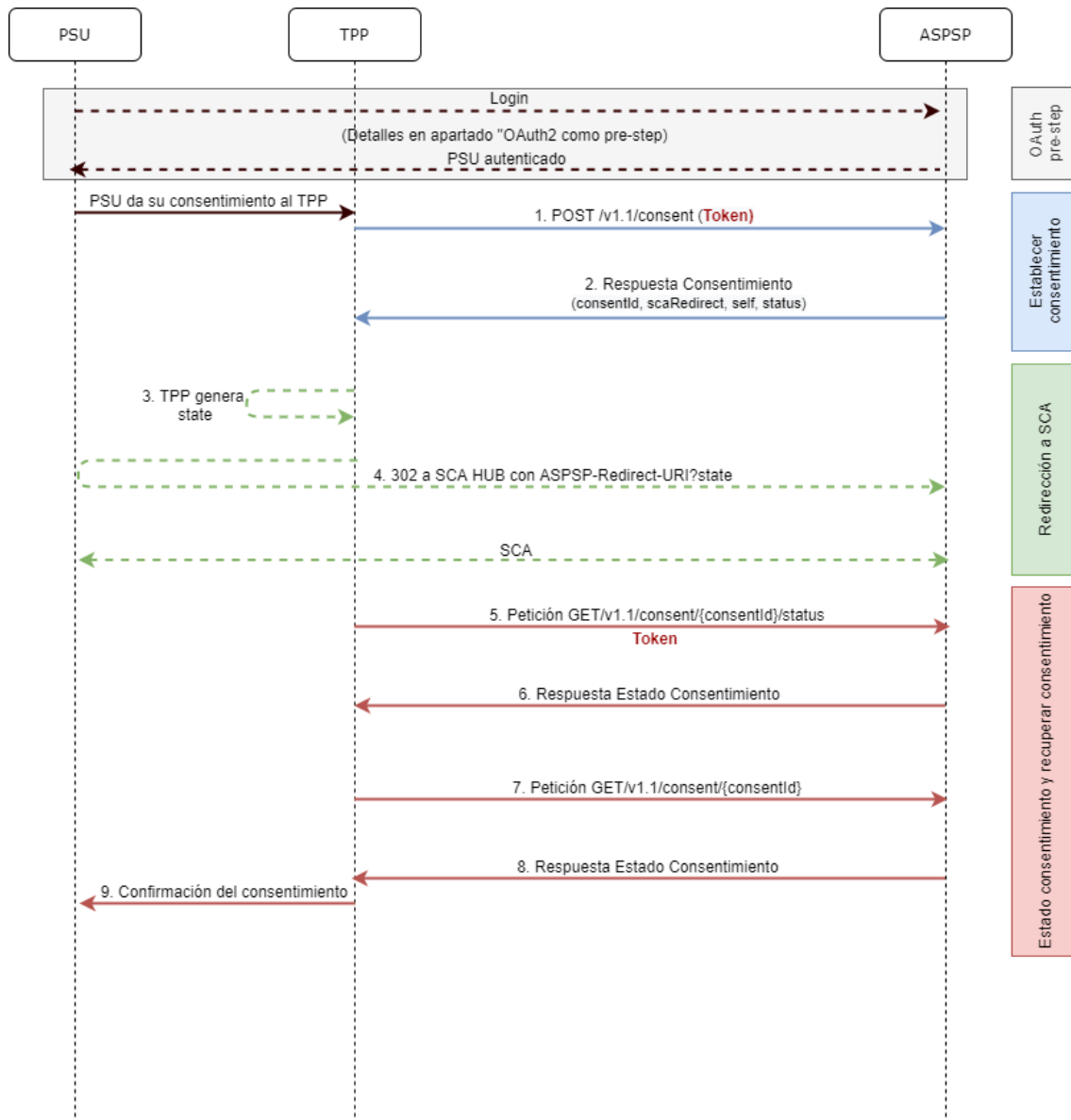


Figure 5: SCA flow by redirection: implicit start of the authorisation process

OAuth2 (pre-step)

The main purpose of this flow is to authenticate the PSU to get access to the services displayed by its ASPSP through the use of an access token obtained after the application of this protocol.

In order to simplify, the detail of this flow has been omitted from the Figure 4 Figure 5: SCA flow by redirection: implicit start of the authorisation process and can be found in the section 6.16.1 OAuth2 as a pre-step.VERIFY

Note: this step is optional. Only applies if no valid access token is available.

PSU gives its consent to the TPP

The PSU gives its consent to the TPP to access its accounts

1. Consent Request (TPP → ASPSP)

The TPP sends a POST request for consent of tokenised account information to the ASPSP. Among the data reported by the TPP are:

- **TPP data:** identifier, name, roles, NCA, certificate ...
- **Consent data:** list of IBANs and/or PANs and types of access to which the PSU has given its consent, recurrence, validity, frequency of daily access...
- **X-Request-ID:** identifier of the operation assigned by the TPP.
- **Access token** from the TPP
- **TPP-Redirect-Preferred:** true - SCA flow preference by redirect
- **TPP-Redirect-URI:** Return URI of the TPP after redirection to SCA.
- **TPP-Explicit-Authorisation-Preferred:** false - TPP preference to initiate authorisation implicitly (current flow)
- **Other data**

2. Response of Consent (ASPSP → TPP)

The ASPSP responds to the TPP indicating that Strong Authentication (SCA) is required by redirecting to the Hub's authentication endpoint, returning:

- **consentStatus-** State of the consent resource.
- **consentId:** identifier generated by the Hub that refers to the consent resource.
- **_links**
 - **scaRedirect:** links to the Hub endpoint where after receiving the redirect from the TPP it redirects back to the scaRedirect of the ASPSP. This URL can add security parameters to allow session maintenance during redirection.

Ex: `https://hub.example.com/auth`
 - **self:** link to the resource generated by the Hub for the request for consent received from the TPP.
 - **state:** link of the Hub to which the TPP will be able to make a consent state query request.

- Other data

3. TPP generates state

The TPP, after receiving the response, generates a value for *state* (XSRF token) to be linked to the PSU browser session.

4. Redirect to scaRedirect (TPP → ASPSP)

The TPP redirects the PSU to the authentication endpoint by adding to it the field *state* as a query-param.

```
HTTP/1.1 302 Found
Location: https://hub.example.com/auth?state=qwerty
```

SCA entre PSU ↔ ASPSP

During this redirection process, the ASPSP will be able to:

- Show ASPSP-PSU interface for SCA

Note: if the consent request does not indicate the accounts for which consent is to be sought, the PSU will be shown its accounts during the SCA process to select which accounts and types of access, it wants to grant to the TPP.

5. State Consent Request (TPP → ASPSP)

The TPP will send a payment state request with *token* to the ASPSP to know the payment state.

6. Response State Consent (ASPSP → TPP)

The ASPSP updates the consent state and responds to the TPP.

7. Request to Retrieve Consent (TPP → ASPSP)

In case the consent request has travelled without indicating the accounts to be given access and the selection of these accounts has been made by the PSU in the ASPSP interface displayed during the redirect of the SCA flow, the TPP will make a request to retrieve information from the consent requested in order to know which accounts have been authorised by the PSU.

The TPP will send a request to the Hub to retrieve consent with the consent identifier provided by the Hub in the consent request response and with a valid access token.

The HUB will make a request to retrieve consent with the consent identifier provided by the ASPSP in the consent request response and with the access token to the ASPSP and, after obtaining a response from the ASPSP, it will send the consent to the TPP.

8. Response Retrieve Consent (ASPSP → TPP)

The ASPSP sends the consent it requested to the TPP along with the accounts and types of access granted to it.

6.4.2.2 SCA flow by redirection: explicit start of authorization process.

Similar to 6.3.1.2 SCA flow by redirection: explicit start of authorization

6.4.3 Payment account information consent

With this service, a TPP, through the Hub, can inform a payment account information consent that is going to access the PSU. This request may or may not be for the specified accounts.

For this reason, the consent request has these variants:

- Establish consent of account information on specified accounts
- Establish consent of account information to obtain a list of all available accounts
- Establish account information consent without indicating accounts
- Establish account information consent to obtain access to all accounts for all PSD2 AIS access types: "accounts", "balances" and / or "transactions"

Note: each consent information will generate a new resource, that is, a new consentId.

6.4.3.1 Request

Endpoint

POST {provider}/{aspsp}/v1.1/consent

Path

Field	Description	Type	Mandat.	Format
-------	-------------	------	---------	--------

PSD2 - APIs Implementation Guide v1.1 for TPPs

provider	URL of the HUB where the service is released.	String	MA	Ex: www.hub.com
aspsp	Name of the ASPSP to which the request is to be made.	String	MA	Ex: aspsp-name

Query parameters:

No additional fields are specified.

Header

Field	Description	Type	Mandate	Format
X-Request-ID	Unique identifier of the operation assigned by the TPP.	String	MA	UUID ^[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}\$ Ex: X-Request-ID: 1b3ab8e8-0fd5-43d2-946e-d75958b172e7
Authorization	Bearer Token. Obtained in a previous authentication on OAuth2.	String	MA	Ex: Authorization: Bearer 2YotnFZFEjr1zCsicMWpAA
PSU-IP-Address	IP address of the HTTP request between the PSU and the TPP.	String	MA	^[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\$ Ex: PSU-IP-Address: 192.168.16.5
PSU-ID	Identifier that the PSU uses to identify itself in its ASPSP.	String	OP	Ex: PSU-ID: 12345678W

PSD2 - APIs Implementation Guide v1.1 for TPPs

	It can be reported even if an OAuth token is being used and, in such a case, the ASPSP could check if the PSU-ID and the token match.			
PSU-ID-Type	Type of the PSU-ID. Necessary in scenarios where the PSU has several PSU-IDs as access possibilities.	String	OP	Ex: PSU-ID-Type: NIF
PSU-Corporate-ID	Identifier of "company" in Online Channels.	String	OP	Ex: PSU-Corporate-ID: user@corporate.com
PSU-Corporate-ID-Type	Type of the PSU-Corporate-ID required by the ASPSP to identify its content.	String	OP	Ex: PSU-Corporate-ID-Type: email
TPP-Redirect-Preferred	<p>If "true", the TPP has communicated to the HUB that it prefers SCA by redirection.</p> <p>If "false", the TPP has informed the HUB that it prefers not to be redirected to SCA and the procedure will be by decoupled flow.</p> <p>If the parameter is not used, the ASPSP will choose the SCA flow to apply depending on the SCA method chosen by the TPP / PSU.</p> <p>EMBEDDED NOT SUPPORTED IN THIS VERSION</p>	Boolean	OP	Ex: TPP-Redirect-Preferred: true
TPP-Redirect-URI	URI of the TPP where the transaction flow must be redirected after any of the SCA phases.	String	COND	<p>^.{1,250}\$</p> <p>Ex: TPP-Redirect-URI: "https://tpp.example.es/cb"</p>

PSD2 - APIs Implementation Guide v1.1 for TPPs

	<p>It is recommended to always use this header field.</p> <p>In the future, this field could change to mandatory.</p>			
TPP-Nok-Redirect-URI	<p>If this URI is contained, the TPP is requesting to redirect the transaction flow to this address instead of the TPP-Redirect-URI in case of a negative result of the SCA method by redirection.</p>	String	OP	<p>^.{12,50}\$</p> <p>Ex: TPP-Nok-Redirect-URI:"https://tpp.example.es/cb/nok"</p>
TPP-Explicit-Authorisation-Preferred	<p>If equal to true, the TPP chooses to initiate the authorisation process separately, e.g. due to the need for authorisation of a set of operations simultaneously.</p> <p>If it is false or the parameter is not used, there is no TPP preference. The TPP takes a direct authorisation of the transaction in the next step.</p> <p>Note: ASPSP might not take it into account if it doesn't support it.</p>	Boolean	OP	<p>Ex: TPP-Explicit-Authorisation-Preferred: false</p>

PSD2 - APIs Implementation Guide v1.1 for TPPs

TPP-Brand-Logging-Information	This field could be used by the TPP to inform the ASPSP about the brand used by the TPP for the PSU. This information can be used to improve communication between the ASPSP and the PSU or the ASPSP and the TPP.	String	OP	$\wedge.\{1,70\}\$$ Ex: TPP-Brand-Logging-Information: TPP Brand
--------------------------------------	--	--------	----	--

Body

Field	Description	Type	Mandant.	Format
access	Requested accesses to services. Only the sub-attributes with tags "accounts", "balances" and "transactions" are accepted. Additionally, the ASPSP can support the "availableAccounts", "availableAccountsWithBalance" or "allPsd2" sub-attributes with value "allAccounts".	Account Access	MA	Ex: "access": {...}
recurringIndicator	Possible values: <ul style="list-style-type: none"> true: recurring access to the account. false: single access. 	Boolean	MA	Ex: "recurringIndicator": true
validUntil	Date until which the consent requests access. To create the consent with the maximum possible access time, the value: 9999-12-31 should be used	String	MA	ISODate Ex: "validUntil": "2018-05-17"

	When consent is recovered, the maximum possible date will be adjusted.			
frequencyPerDay	Indicates the frequency of access to the account per day. 1 for single use.	Integer	MA	Ex: "frequencyPerDay": 4
combinedServiceIndicator	The session support is specified by the access token. The value of this field will be ignored by the ASPSP.	Boolean	MA	Ex: "combinedServiceIndicator": false

6.4.3.2 Response

HTTP Code

201 if the resource has been created

Header

Field	Description	Type	Mandate	Format
Location	Contains the hyperlink to the generated resource	String	MA	Max512Text Ex: Location: /v1.1/consents/[consentId]
X-Request-ID	Unique identifier of the operation assigned by the TPP.	String	MA	UUID ^[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}\$ Ex: X-Request-ID: 1b3ab8e8-0fd5-43d2-946e-d75958b172e7

ASPSP-SCA-Approach	<p>Value returned if the SCA method has been set. Possible values:</p> <ul style="list-style-type: none"> • EMBEDDED • DECOUPLED • REDIRECT <p>The OAuth based SCA will be taken as REDIRECT.</p>	String	COND	Ex: ASPSP-SCA-Approach: REDIRECT
---------------------------	--	--------	------	----------------------------------

Body

Field	Description	Type	Mandat.	Format
consentStatus	Consent authentication state. Defined values in 9.5 Consent states	String	MA	Ex: "consentStatus": "received"
consentId	Identifier of the resource that refers to the consent. It must be contained if consent was generated.	String	MA	^.{1,36}\$ Ex: "consentId": "123-QWE-456"
_links	<p>List of hyperlinks to be recognized by the TPP. Supported types in this response:</p> <ul style="list-style-type: none"> • scaRedirect: in case of SCA by redirection. Link where the PSU browser must be redirected by the TPP. • startAuthorisation: in case an explicit start of transaction authorization is required (no SCA method selection) • self: link to the resource created by this request. 	Links	MA	Ex: "_links": {...}

PSD2 - APIs Implementation Guide v1.1 for TPPs

	<ul style="list-style-type: none"> state: link to retrieve the state of the transaction. scaStatus: link to consult the SCA state corresponding to the authorisation sub-resource. This link is only contained if an authorization sub-resource has been created. 			
psuMessage	Text to show to the PSU.	String	OP	$\wedge.\{1,500\}$ \$ Ex: "psuMessage": "Información para PSU"
tppMessages	Message to the TPP	List<Tp pMessage>	OP	Ex: "tppMessages": [...]

6.4.3.3 Examples

Example request consent on accounts specified with SCA by redirection

POST <https://www.hub.com/aspsp-name/v1.1/consents>

Content-Encoding: gzip

Content-Type: application/json

X-Request-ID: 10391c7e-ad88-49ec-a2ad-00aacb1f6541

Authorization: Bearer 2YotnFZFEjrlzCsicMWpAA

PSU-IP-Address: 192.168.8.16

PSU-IP-Port: 443

PSU-Accept: application/json

PSU-Accept-Charset: utf-8

PSU-Accept-Encoding: gzip

PSU-Accept-Language: es-ES

PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0) Gecko/20100101 Firefox/54.0

PSU-Http-Method: POST

PSD2 - APIs Implementation Guide v1.1 for TPPs

```
PSU-Device-ID: f8b3feda-6fe3-11e8-adc0-fa7ae01bbebc
PSU-GEO-Location: GEO:12.526347;54.649862
TPP-Redirect-Preferred: true
TPP-Redirect-URI: https://www.tpp.com/cb
TPP-Nok-Redirect-URI: https://www.tpp.com/cb/nok
Date: Sun, 26 Sep 2017 15:02:37 GMT
{
  "access": {
    "balances": [
      {
        "iban": "ES1111111111111111111111"
      },
      {
        "iban": "ES2222222222222222222222",
        "currency": "USD"
      },
      {
        "iban": "ES3333333333333333333333"
      }
    ],
    "transactions": [
      {
        "iban": "ES1111111111111111111111"
      }
    ]
  },
  "recurringIndicator": true,
  "validUntil": "2018-05-17",
  "frequencyPerDay": 4
}
```

Example of a consent request for a list of available accounts with SCA by redirection

```
POST https://www.hub.com/aspsp-name/v1.1/consent
Content-Encoding: gzip
Content-Type: application/json
X-Request-ID: 10391c7e-ad88-49ec-a2ad-00aacb1f6541
Authorization: Bearer 2YotnFZFEjrlzCsicMWpAA
PSU-IP-Address: 192.168.8.16
PSU-IP-Port: 443
```

PSD2 - APIs Implementation Guide v1.1 for TPPs

```
PSU-Accept: application/json
PSU-Accept-Charset: utf-8
PSU-Accept-Encoding: gzip
PSU-Accept-Language: es-ES
PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0
PSU-Http-Method: POST
PSU-Device-ID: f8b3feda-6fe3-11e8-adc0-fa7ae01bbebc
PSU-GEO-Location: GEO:12.526347;54.649862
TPP-Redirect-Preferred: true
TPP-Redirect-URI: https://www.tpp.com/cb
TPP-Nok-Redirect-URI: https://www.tpp.com/cb/nok
Date: Sun, 26 Sep 2017 15:02:37 GMT
{
  "access": {
    "availableAccounts": "allAccounts"
  },
  "recurringIndicator": false,
  "validUntil": "2018-05-17",
  "frequencyPerDay": 1
}
```

Example response in case of SCA by redirection with implicitly generated authorisation sub-resource

```
HTTP/1.1 201 Created
X-Request-ID: 10391c7e-ad88-49ec-a2ad-00aacb1f6541
ASPSP-SCA-Approach: REDIRECT
Date: Sun, 26 Sep 2017 15:02:43 GMT
Location: /v1.1/consents/123-asdf-456
Content-Type: application/json
{
```

```
  "consentStatus": "received",
  "consentId": "123-asdf-456",
  "_links": {
    "scaRedirect": {
```

PSD2 - APIs Implementation Guide v1.1 for TPPs

```
        "href": "https://hub.example.es/authorize "
      },
      "self": {
        "href": "/v1.1/consents/123-asdf-456",
      },
      "state": {
        "href": "/v1.1/consents/123-asdf-456/state"
      },
      "scaStatus": {
        "href": "/v1.1/consents/123-asdf-456/authorisations/123auth456"
      }
    }
  }
}
```

6.4.4 Obtain consent state

This service allows the TPP to know the state of a previously initiated consent request.

6.4.4.1 Request

Endpoint

GET {provider}/{aspsp}/v1.1/consents/{consent-id}/state

Path

Field	Description	Type	Mandat.	Format
provider	URL of the HUB where the service is released.	String	MA	Ex: www.hub.com
aspsp	Name of the ASPSP to which the request is to be made.	String	MA	Ex: aspsp-name
consentId	Identifier of the resource that refers to the consent.	String	MA	^. {1,36}\$ Ex:123-qwerty-456

	Previously sent in response to a consent request message from the TPP to the HUB.			
--	---	--	--	--

Query parameters:

No additional fields are specified.

Header

Field	Description	Type	Mandant.	Format
X-Request-ID	Unique identifier of the request assigned by the TPP.	String	MA	UUID ^[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}\$ Ex: X-Request-ID: 1b3ab8e8-0fd5-43d2-946e-d75958b172e7
Authorization	Bearer Token. Obtained in a previous authentication on OAuth2.	String	MA	Ex: Authorization: Bearer 2YotnFZFEjr1zCsicMWpAA

Body

No additional data travels.

6.4.4.2 Response

HTTP Code

200 if the request has been successful.

This message is returned by the HUB to the TPP in response to the consent state request message.

Header

Field	Description	Type	Mandat.	Format
X-Request-ID	Unique identifier of the request assigned by the TPP.	String	MA	UUID $\wedge[0-9a-fA-F]\{8\}-[0-9a-fA-F]\{4\}-[0-9a-fA-F]\{4\}-[0-9a-fA-F]\{4\}-[0-9a-fA-F]\{12\}\$$ Ex: X-Request-ID: 1b3ab8e8-0fd5-43d2-946e-d75958b172e7

Body

Field	Description	Type	Mandat.	Format
consentStatus	Consent authentication state. Defined values in 9.5 Consent states	String	MA	Ex: "consentStatus": "valid"
psuMessage	Text to show to the PSU	String	OP	$\wedge.\{1,500\}\$$ Ex: "psuMessage": "Información para PSU"
tppMessages	Message to the TPP	List<Tp pMessage>	OP	Ex: "tppMessages": [...]

6.4.4.3 Examples

Example of request

GET <https://www.hub.com/aspsp-name/v1.1/consents/123asdf456/state>

Accept: application/json

X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc

Authorization: Bearer 2YotnFZFEjrlzCsicMWpAA

PSU-IP-Address: 192.168.8.16

PSU-IP-Port: 443

PSD2 - APIs Implementation Guide v1.1 for TPPs

```
PSU-Accept: application/json
PSU-Accept-Charset: utf-8
PSU-Accept-Encoding: gzip
PSU-Accept-Language: es-ES
PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0
PSU-Http-Method: GET
PSU-Device-ID: f8b3feda-6fe3-11e8-adc0-fa7ae01bbebc
PSU-GEO-Location: GEO:12.526347;54.649862
Date: Sun, 26 Sep 2017 15:02:48 GMT
```

Example response

```
HTTP/1.1 200 Ok
X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc
Date: Sun, 26 Sep 2017 15:02:50 GMT
Content-Type: application/json
{
    "consentStatus": "valid"
}
```

6.4.5 Retrieve consent information

6.4.5.1 Request

This message is sent by the TPP to the HUB as a request to retrieve the information from a previously created consent.

Endpoint

GET {provider}/{aspsp}/v1.1/consents/{consentId}

Path

Field	Description	Type	Mandat.	Format
provider	URL of the HUB where the service is released.	String	MA	Ex: www.hub.com

PSD2 - APIs Implementation Guide v1.1 for TPPs

aspsp	Name of the ASPSP to which the request is to be made.	String	MA	Ex: aspsp-name
consentId	Identifier of the resource that refers to the consent. Previously sent in response to a consent request message from the TPP to the HUB.	String	MA	^.{1,36}\$ Ex: 7890-asdf-4321

Query parameters:

No additional fields are specified.

Header

The same as those defined in the section 6.4.4.1

Body

No additional data travels.

6.4.5.2 Response

HTTP Code

200 if the request has been successful.

This message is returned by the HUB to the TPP in response to the consent information retrieval message.

Header

The same as those defined in the section 6.4.4.2

Body

Field	Description	Type	Mand at.	Format
-------	-------------	------	----------	--------

PSD2 - APIs Implementation Guide v1.1 for TPPs

access	Requested accesses to services. Only the sub-attributes with tags "accounts", "balances" and "transactions" are accepted. Additionally, the ASPSP can support the "availableAccounts", "availableAccountsWithBalance" or "allPsd2" sub-attributes with value "allAccounts".	AccountAccesses	MA	Ex: "access": {...}
recurringIndicator	Possible values: <ul style="list-style-type: none"> true: recurring access to the account. false: single access. 	Boolean	MA	Ex: "recurringIndicator": true
validUntil	Date until which the consent requests access. To create the consent with the maximum possible access time, the value: 9999-12-31 should be used When consent is recovered, the maximum possible date will be adjusted.	String	MA	ISODate Ex: "validUntil": "2018-05-17"
frequencyPerDay	Indicates the frequency of access to the account per day. 1 if single-access.	Integer	MA	Ex: "frequencyPerDay": 4
lastActionDate	Date of the last modification made to the consent.	String	MA	ISODate Ex: "lastActionDate": "2018-01-01"
consentStatus	Consent authentication state. Values defined in annexes.	String	MA	Ex: "consentStatus": "valid"

PSD2 - APIs Implementation Guide v1.1 for TPPs

psuMessage	Text to show to the PSU	String	OP	^.{1,500} \$ Ex: "psuMessage": "Información para PSU"
tppMessages	Message to the TPP	List<TppMessage>	OP	Ex: "tppMessages": [...]

6.4.5.3 Examples

Example of request

```
GET https://www.hub.com/aspsp-name/v1.1/consents/7890-asdf-4321/
Accept: application/json
X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc
Authorization: Bearer 2YotnFZFEjrlzCsicMWpAA
PSU-IP-Address: 192.168.8.16
PSU-IP-Port: 443
PSU-Accept: application/json
PSU-Accept-Charset: utf-8
PSU-Accept-Encoding: gzip
PSU-Accept-Language: es-ES
PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0
PSU-Http-Method: GET
PSU-Device-ID: f8b3feda-6fe3-11e8-adc0-fa7ae01bbebc
PSU-GEO-Location: GEO:12.526347;54.649862
Date: Sun, 26 Sep 2017 15:02:48 GMT
```

Example response on consent with specified accounts

```
HTTP/1.1 200 Ok
X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc
Date: Sun, 26 Sep 2017 15:02:50 GMT
Content-Type: application/json
{
  "access": {
    "balances": [
      {
```

PSD2 - APIs Implementation Guide v1.1 for TPPs

```

        "iban": "ES1111111111111111111111111111",
    },
    {
        "iban": "ES2222222222222222222222222222",
        "currency": "USD"
    },
    {
        "iban": "ES3333333333333333333333333333"
    }
],
"transactions": [
    {
        "iban": "ES1111111111111111111111111111"
    }
]
},
"recurringIndicator": true,
"validUntil": "2018-05-17",
"frequencyPerDay": 4,
"lastActionDate": "2018-01-17",
"consentStatus": "valid"
}

```

Example response on global consent availableAccounts

HTTP/1.1 200 OK

X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc

Date: Sun, 26 Sep 2017 15:02:50 GMT

Content-Type: application/json

```
{
  "access": {
    "availableAccounts": "allAccounts"
  },
  "recurringIndicator": true,
  "validUntil": "2018-05-17",
  "frequencyPerDay": 4,
  "lastActionDate": "2018-01-17",
  "consentStatus": "valid"
}
```

6.4.6 Remove consent

6.4.6.1 Request

This request can be sent by a TPP to the HUB to request the removal of a previously created consent.

Endpoint

DELETE {provider}/{aspsp}/v1.1/consents/{consentId}

Path

Field	Description	Type	Mandate.	Format
provider	URL of the HUB where the service is released.	String	MA	Ex: www.hub.com
aspsp	Name of the ASPSP to which the request is to be made.	String	MA	Ex: aspsp-name
consentId	Identifier of the resource that refers to the consent. Previously sent in response to a consent request message from the TPP to the HUB.	String	MA	^.{1,36}\$ Ex: 7890-asdf-4321

Query parameters:

No additional fields are specified.

Header

The same as those defined in the section 6.4.4.1

Body

No additional data travels.

6.4.6.2 Response

HTTP Code

204 if the request has been successful.

This message is sent by the HUB to the TPP in response to the request to remove consent.

Header

The same as those defined in the section 6.4.4.2

Body

No additional fields are specified.

6.4.6.3 Examples

Example of request

```
DELETE https://www.hub.com/aspsp-name/v1.1/consents/7890-asdf-4321
Accept: application/json
X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc
Authorization: Bearer 2YotnFZFEjrlzCsicMWpAA
PSU-IP-Address: 192.168.8.16
PSU-IP-Port: 443
PSU-Accept-Charset: utf-8
PSU-Accept-Encoding: gzip
PSU-Accept-Language: es-ES
PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0
PSU-Http-Method: DELETE
PSU-Device-ID: f8b3feda-6fe3-11e8-adc0-fa7ae01bbebc
PSU-GEO-Location: GEO:12.526347;54.649862
Date: Sun, 26 Sep 2017 15:02:48 GMT
```

Example response

```
HTTP / 1.1 204 Ok
X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc
Date: Sun, 26 Sep 2017 15:02:50 GMT
```


6.5 AIS: Account data reading service

6.5.1 Reading list of accounts

This service allows to obtain a list of PSU accounts, including account balances if requested and consent is available.

This request is used both for the list of available accounts and for the list of account details. Depending on the consent used in the request.

As a prerequisite, it is assumed that the PSU has given its consent to this access and has been stored by the ASPSP.

Operation of the service according to the type of access specified in the consent:

Type of access	Description
availableAccounts	<p>This type of access is associated with single-use consent. If the consent associated with the request has this type of access, it will be a one-time consent and it will be possible to obtain:</p> <ul style="list-style-type: none"> List of all available PSU accounts. <p>You will not be able to obtain:</p> <ul style="list-style-type: none"> Account balances (unless supported by ASPSP) Links to balance or transaction endpoints
availableAccountsWithBalance	<p>This type of access is associated with single-use consent. If the consent associated with the request has this type of access, it will be a one-time consent and it will be possible to obtain:</p> <ul style="list-style-type: none"> List of all available PSU accounts. Account balances (unless supported by ASPSP) <p>You will not be able to obtain:</p> <ul style="list-style-type: none"> Links to balance or transaction endpoints
account	<p>If the consent associated with the request has this type of access, the accounts included in the consent with access type "account" may be listed.</p>
balances	<p>If the consent associated with the request has this type of access, the accounts included in the consent with the access type "balances" may be listed and their balances may be obtained if the ASPSP supports it.</p>

transactions	If the consent has accounts with this type of access, these accounts may be listed with the access type "account". This type of access does not imply a "balance" type of access.
allPsd2	If the consent associated with the request has this type of access, the accounts included in the consent may be listed and their balances may be obtained. Note: allPsd2 grants all three types of access.

6.5.1.1 Request

Endpoint

GET {provider}/{aspsp}/v1.1/accounts{query-parameters}

Path

Field	Description	Type	Mand at.	Format
provider	URL of the HUB where the service is released	String	MA	Ex: www.hub.com
aspsp	Name of the ASPSP to which the request is to be made.	String	MA	Ex: aspsp-name

Query parameters:

Field	Description	Type	Mand at.	Format
withBalance	If included, this function includes balances. This request will be rejected if the access to balances is not covered by the consent or the ASPSP does not support this parameter.	Boolean	OP	Ex: true

Header

Field	Description	Type	Mand at.	Format
X-Request-ID	Unique identifier of the operation assigned by the TPP.	String	MA	UUID

				$^{\wedge}[0-9a-fA-F]\{8\}-[0-9a-fA-F]\{4\}-[0-9a-fA-F]\{4\}-[0-9a-fA-F]\{4\}-[0-9a-fA-F]\{12\}\$$ Ex: X-Request-ID: 1b3ab8e8-0fd5-43d2-946e-d75958b172e7
Authorization	Bearer Token. Obtained in a previous authentication on OAuth2.	String	MA	Ex: Authorization: Bearer 2YotnFZFEjr1zCsicMWpAA
Consent-ID	Identifier of the consent obtained in the transaction to request consent.	String	MA	$^{\wedge}\{1,36\}\$$ Ex: Consent-ID: 7890-asdf-4321
PSU-IP-Address	IP address of the HTTP request between the PSU and the TPP. It should only be included if this request was actively initiated by the PSU.	String	COND	$^{\wedge}[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}\$$ Ex: PSU-IP-Address: 192.168.16.5

Body

No data travels in the body of this request.

6.5.1.2 Response

Header

Field	Description	Type	Mandant.	Format
X-Request-ID	Unique identifier of the operation assigned by the TPP.	String	MA	UUID

PSD2 - APIs Implementation Guide v1.1 for TPPs

				<p><code>^[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}\$</code></p> <p>Ex:</p> <p>X-Request-ID: 1b3ab8e8-0fd5-43d2-946e-d75958b172e7</p>
--	--	--	--	--

Body

Field	Description	Type	Manda t.	Format
accounts	List of available accounts.	List<AccountDetails>	MA	Ex: "accounts": []
psuMessage	Text to show to the PSU.	String	OP	<code>^.{1,500}\$</code> Ex: "psuMessage": "Información para PSU"
tppMessages	Message to the TPP	List<TppMessage>	OP	Ex: "tppMessages": [...]

6.5.1.3 Examples

Example request to obtain a list of accounts accessible from the PSU

GET <https://www.hub.com/aspsp-name/v1.1/accounts>
Content-Encoding: gzip
Content-Type: application/json
X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc
Authorization: Bearer 2YotnFZFEjrlzCsicMWpAA
Consent-ID: 7890-asdf-4321
PSU-IP-Address: 192.168.8.16
PSU-IP-Port: 443
PSU-Accept: application/json
PSU-Accept-Charset: utf-8
PSU-Accept-Encoding: gzip

PSD2 - APIs Implementation Guide v1.1 for TPPs

```
PSU-Accept-Language: es-ES
PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0
PSU-Http-Method: GET
PSU-Device-ID: f8b3feda-6fe3-11e8-adc0-fa7ae01bbebc
PSU-GEO-Location: GEO:12.526347;54.649862
Date: Sun, 26 Sep 2017 15:02:48 GMT
```

Example response to obtain a list of accessible PSU accounts

Response where consent has been given on two different IBANs.

```
HTTP/1.1 200 Ok
X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc
Date: Sun, 26 Sep 2017 15:02:50 GMT
Content-Type: application/json
{
  "accounts": [
    {
      "resourceId": "3dc3d5b3-7023-4848-9853-f5400a64e80f",
      "iban": "ES1111111111111111111111",
      "currency": "EUR",
      "product": "Girokonto",
      "cashAccountType": "CACC",
      "name": "Main Account",
      "_links": {
        "balances": {
          "href": "/v1.1/accounts/3dc3d5b3-7023-4848-9853-f5400a64e80f/balances"
        },
        "transactions": {
          "href": "/v1.1/accounts/3dc3d5b3-7023-4848-9853-f5400a64e80f/transactions"
        }
      }
    },
    {

```

```

    "resourceId": "3dc3d5b3-7023-4848-9853-f5400a64e81g",
    "iban": "ES2222222222222222222222",
    "currency": "USD",
    "cashAccountType": "CACC",
    "name": "US Dollar Account",
    "_links": {
      "balances": {
        "href": "/v1.1/accounts/3dc3d5b3-7023-4848-9853-f5400a64e81g/balances"
      }
    }
  }
]
}

```

6.5.2 Reading account details

This service allows you to read the details of an account with the balances if they are required.

As a prerequisite, it is assumed that the PSU has given its consent to this access and has been stored by the ASPSP.

Operation of the service according to the type of access specified in the consent:

Type of access	Description
availableAccounts	With this type of access, it is not possible to use this service.
availableAccountsWithBalance	With this type of access, it is not possible to use this service.
account	If the consent associated with the request has this type of access, the account can be consulted.
balances	If the consent associated with the request has this type of access, the account can be consulted, and its balances can be obtained if the ASPSP supports it.

transactions	If the consent has accounts with this type of access, the account can be consulted with the access type "account". This type of access does not imply a "balance" type of access.
allPsd2	If the consent associated with the request has this type of access, the account can be consulted, and its balances can be obtained. Note: allPsd2 grants all three types of access.

6.5.2.1 Request

Endpoint

GET {provider}/{aspsp}/v1.1/accounts/{account-id}{query-parameters}

Path

Field	Description	Type	Mandat.	Format
provider	URL of the HUB where the service is released	String	MA	Ex: www.hub.com
aspsp	Name of the ASPSP to which the request is to be made.	String	MA	Ex: aspsp-name
account-id	Account identifier assigned by ASPSP	String	MA	^{1,100} \$ Ex: account-id = a1q5w

Query parameters:

Field	Description	Type	Mandat.	Format
withBalance	If included, this function includes balances. This request will be rejected if the access to balances is not covered by the consent or the ASPSP does not support this parameter.	Boolean	OP	Ex: true

Header

The same as those defined in the section 6.5.1.1

Body

No data travels in the body of this request.

6.5.2.2 Response

HTTP Code

200 if the request has been successful.

Header

The same as those defined in the section 6.5.1.2

Body

Field	Description	Type	Mandat.	Format
account	Detailed account information	Account Details	MA	Ex: "account": {...}
psuMessage	Text to show to the PSU	String	OP	^.{1,500} \$ Ex: "psuMessage": "Información para PSU"
tppMessages	Message to the TPP	List<Tpp Message>	OP	Ex: "tppMessages": [...]

6.5.2.3 Examples

Example of request

GET <https://www.hub.com/aspsp-name/v1.1/accounts/3dc3d5b3-7023-4848-9853-f5400a64e80f>

Content-Encoding: gzip

Content-Type: application/json

X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc

Authorization: Bearer 2YotnFZFEjrlzCsicMWpAA

Consent-ID: 7890-asdf-4321

PSD2 - APIs Implementation Guide v1.1 for TPPs

PSU-IP-Address: 192.168.8.16
PSU-IP-Port: 443
PSU-Accept: application/json
PSU-Accept-Charset: utf-8
PSU-Accept-Encoding: gzip
PSU-Accept-Language: es-ES
PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0
PSU-Http-Method: GET
PSU-Device-ID: f8b3feda-6fe3-11e8-adc0-fa7ae01bbebc
PSU-GEO-Location: GEO:12.526347;54.649862
Date: Sun, 26 Sep 2017 15:02:48 GMT

Example response

HTTP/1.1 200 Ok
X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc
Date: Sun, 26 Sep 2017 15:02:50 GMT
Content-Type: application/json
{
 "account": {
 "resourceId": "3dc3d5b3-7023-4848-9853-f5400a64e80f",
 "iban": "ES1111111111111111111111",
 "currency": "EUR",
 "ownerName": "Heike Mustermann",
 "product": "Girokonto",
 "cashAccountType": "CACC",
 "name": "Main Account",
 "_links": {
 "balances": {
 "href": "/v1.1/accounts/3dc3d5b3-7023-4848-9853-f5400a64e80f/balances"
 },
 "transactions": {
 "href": "/v1.1/accounts/3dc3d5b3-7023-4848-9853-f5400a64e80f/transactions"
 }
 }
 }
}

```

    }
  }
}

```

6.5.3 Balance reading

This service allows you to obtain the balances of an account determined by its identifier.

As a prerequisite, it is assumed that the PSU has given its consent to this access and has been stored by the ASPSP.

Operation of the service according to the type of access indicated in the consent:

Type of access	Description
availableAccounts	With this type of access, it is not possible to use this service.
availableAccountsWithBalance	With this type of access, it is not possible to use this service.
account	With this type of access, it is not possible to use this service.
balances	If the consent associated with the request has this type of access, the account balances may be consulted.
transactions	With this type of access, it is not possible to use this service.
allPsd2	If the consent associated with the request has this type of access, the account balances may be consulted. Note: allPsd2 grants all three types of access.

6.5.3.1 Request

Endpoint

GET {provider}/{aspsp}/v1.1/accounts/{account-id}/balances

Path

Field	Description	Type	Mandat.	Format
provider	URL of the HUB where the service is released	String	MA	Ex: www.hub.com
aspsp	Name of the ASPSP to which the request is to be made.	String	MA	Ex: aspsp-name
account-id	Identifier of the account to be used when reading data. Obtained previously in the reading of the list of accounts. It must be valid, at least, for the duration of the consent. This id can be tokenized.	String	MA	^.{1,100} \$ Ex: account-id = a1q5w

Query parameters:

No additional fields are specified.

Header

The same as those defined in the section 6.5.1.1

Body

No data travels in the body of this request.

6.5.3.2 Response

HTTP Code

200 if the request has been successful.

Header

The same as those defined in the section 6.5.1.2

Body

Field	Description	Type	Mandat.	Format
account	Identifier of the account being queried.	AccountReference	OP	Ex: "account": {...}

PSD2 - APIs Implementation Guide v1.1 for TPPs

	Note: it is recommended to use it as it could become a mandatory parameter in future versions.			
balances	A list of balances with respect to an account.	List<Balance>	MA	Ex: "balances": {...}
psuMessage	Text to show to the PSU.	String	OP	^.{1,500} \$ Ex: "psuMessage" : "Información para PSU"
tppMessages	Message to the TPP	List<TppMessage>	OP	Ex: "tppMessages" : [...]

6.5.3.3 Examples

Example of request

GET <https://www.hub.com/aspsp-name/v1.1/accounts/3dc3d5b3-7023-4848-9853-f5400a64e81g/balances>

Accept: application/json

X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc

Authorization: Bearer 2YotnFZFEjrlzCsicMWpAA

Consent-ID: 7890-asdf-4321

PSU-IP-Address: 192.168.8.16

PSU-IP-Port: 443

PSU-Accept: application/json

PSU-Accept-Charset: utf-8

PSU-Accept-Encoding: gzip

PSU-Accept-Language: es-ES

PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0) Gecko/20100101 Firefox/54.0

PSU-Http-Method: GET

PSU-Device-ID: f8b3feda-6fe3-11e8-adc0-fa7ae01bbebc

PSU-GEO-Location: GEO:12.526347;54.649862

Date: Sun, 26 Sep 2017 15:02:48 GMT

Example of response

HTTP/1.1 200 Ok

X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc

Date: Sun, 26 Sep 2017 15:02:50 GMT

Content-Type: application/json

```
{
  "account": {
    "iban": "ES1111111111111111111111"
  },
  "balances": [
    {
      "balanceType": "closingBooked",
      "balanceAmount": {
        "currency": "EUR",
        "amount": "500.00"
      },
      "referenceDate": "2017-10-25"
    },
    {
      "balanceType": "expected",
      "balanceAmount": {
        "currency": "EUR",
        "amount": "900.00"
      },
      "lastChangeDateTime": "2017-10-25T15:30:35.035Z"
    }
  ]
}
```

6.5.4 Reading of transactions

This service allows to obtain the transactions of an account determined by its identifier.

PSD2 - APIs Implementation Guide v1.1 for TPPs

As a prerequisite, it is assumed that the PSU has given its consent to this access and has been stored by the ASPSP.

Operation of the service according to the type of access specified in the consent:

Type of access	Description
availableAccounts	With this type of access, it is not possible to use this service.
availableAccountsWithBalance	With this type of access, it is not possible to use this service.
account	With this type of access, it is not possible to use this service.
balances	If the consent associated with the request has this type of access, it will be allowed to request the balances if the ASPSP supports it.
transactions	If the consent associated with the request provides for this type of access, the movements of the account may be consulted.
allPsd2	If the consent associated with the request has this type of access, the account balances may be consulted. Note: allPsd2 grants all three types of access.

6.5.4.1 Request

Endpoint

GET {provider}/{aspsp}/v1.1/accounts/{account-id}/transactions{query-parameters }

Path

Field	Description	Type	Mandat.	Format
provider	URL of the HUB where the service is released	String	MA	Ex: www.hub.com
aspsp	Name of the ASPSP to which the request is to be made.	String	MA	Ex: aspsp-name

PSD2 - APIs Implementation Guide v1.1 for TPPs

account-id	<p>Identifier of the account to be used when reading data.</p> <p>Obtained previously in the reading of the list of accounts.</p> <p>It must be valid, at least, for the duration of the consent.</p> <p>This id can be tokenized.</p>	String	MA	<p>^.{1,100} \$</p> <p>Ex: account-id = a1q5w</p>
-------------------	--	--------	----	---

Query parameters:

Field	Description	Type	Mandat.	Format
dateFrom	<p>Query start date (including dateFrom). Mandatory if delta access is not required and if "bookingStatus" is not equal to "information". It could be ignored if the delta function or the "information" state is used.</p> <p>For booked transactions, the relevant date is the date of the bookingDate entry. For pending transactions, the relevant date is the entry "entryDate".</p>	String	COND	<p>ISODate</p> <p>Ex: dateFrom=2017-10-25</p>
dateTo	<p>Consultation end date. By default, it is the time of the request if it is not reported. It could be ignored if the delta function is used.</p>	String	OP	<p>ISODate</p> <p>Ex: dateTo=2017-11-05</p>

	For booked transactions, the relevant date is the date of the bookingDate entry. For pending transactions, the relevant date is the entry "entryDate".			
entryReferenceFrom	When specified, it would give us the results from the call with entryReferenceFrom before the one given. If contained, the dateFrom and dateTo attributes are ignored. Note: only if supported by ASPSP.	String	OP	Ex: entryReferenceFrom =1234-asdf-567
bookingStatus	State of returned transactions. Supported values: <ul style="list-style-type: none"> booked (OB) pending (OP) both (OP) Note: pending and both only if they are supported by ASPSP. Additionally, the state is supported: <ul style="list-style-type: none"> Information (OB) To return the list of standing orders.	String	MA	Ex: bookingStatus = booked

Note: in case bookingStatus is equal to "information", the query param dateFrom, dateTo, withBalance will be ignored and have no effect on the result.

The same as those defined in the section 6.5.1.1

Body

No data travels in the body of this request.

6.5.4.2 Response

HTTP Code

200 if the request has been successful.

Header

The same as those defined in the section 6.5.1.2

Body

Field	Description	Type	Manda t.	Format
account	Identifier of the account being queried. Note: it is recommended to use it as it could become a mandatory parameter in future versions.	AccountRefer ence	OP	Ex: "account": {...}
transaction s	Return of data in JSON format, when the returned data are small in length.	AccountRepo rt	OP	Ex: "transactions": {...}
balances	A list of balances with respect to an account.	List<Balance >	OP	Ex: "balances": {...}
psuMessag e	Text to show to the PSU	String	OP	^. {1,500} \$ Ex: "psuMessage": "Información para PSU"
tppMessage s	Message to the TPP	List<TppMes sage>	OP	Ex: "tppMessages": [...]

6.5.4.3 Examples

Example of a search request sending search criteria by dateTo and dateFrom

PSD2 - APIs Implementation Guide v1.1 for TPPs

GET

<https://www.hub.com/aspsp-name/v1.1/accounts/qwer3456tzui7890/transactions?dateFrom=2017-10-25&dateTo=2017-11-05&bookingStatus=both>

Accept: application/json

X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc

Authorization: Bearer 2YotnFZFEjrlzCsicMWpAA

Consent-ID: 7890-asdf-4321

PSU-IP-Address: 192.168.8.16

PSU-IP-Port: 443

PSU-Accept: application/json

PSU-Accept-Charset: utf-8

PSU-Accept-Encoding: gzip

PSU-Accept-Language: es-ES

PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0) Gecko/20100101 Firefox/54.0

PSU-Http-Method: GET

PSU-Device-ID: f8b3feda-6fe3-11e8-adc0-fa7ae01bbebc

PSU-GEO-Location: GEO:12.526347;54.649862

Date: Sun, 26 Sep 2017 15:02:48 GMT

Example of a search request by sending search criterion entryReferenceFrom

GET <https://www.hub.com/aspsp-name/v1.1/accounts/qwer3456tzui7890/transactions?entryReferenceFrom=1234-asd-4564700&bookingStatus=both>

Accept: application/json

X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc

Authorization: Bearer 2YotnFZFEjrlzCsicMWpAA

Consent-ID: 7890-asdf-4321

PSU-IP-Address: 192.168.8.16

PSU-IP-Port: 443

PSU-Accept: application/json

PSU-Accept-Charset: utf-8

PSU-Accept-Encoding: gzip

PSU-Accept-Language: es-ES

PSD2 - APIs Implementation Guide v1.1 for TPPs

```
PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0

PSU-Http-Method: GET

PSU-Device-ID: f8b3feda-6fe3-11e8-adc0-fa7ae01bbebc

PSU-GEO-Location: GEO:12.526347;54.649862

Date: Sun, 26 Sep 2017 15:02:48 GMT
```

Example of a response with page numbering

```
HTTP/1.1 200 Ok
X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc
Date: Sun, 26 Sep 2017 15:02:50 GMT
Content-Type: application/json

{
  "account": {
    "iban": "ES1111111111111111111111"
  },
  "transactions": {
    "booked": [
      {
        "transactionId": "1234567",
        "creditorName": "John Miles",
        "creditorAccount": {
          "iban": "ES1111111111111111111111"
        },
        "transactionAmount": {
          "currency": "EUR",
          "amount": "256.67"
        },
        "bookingDate": "2017-10-25",
        "valueDate": "2017-10-26",
        "remittanceInformationUnstructured": "Example for
Remittance Information"
      },
      {
        "transactionId": "1234568",
        "debtorName": "Paul Simpson"
      }
    ]
  }
}
```

PSD2 - APIs Implementation Guide v1.1 for TPPs

```
      "debtorAccount": {
        "iban": "NL354543123456900"
      },
      "transactionAmount": {
        "currency": "EUR",
        "content": "343.01"
      },
      "bookingDate": "2017-10-25",
      "valueDate": "2017-10-26",
      "remittanceInformationUnstructured": "Another example
for Remittance Information"
    }
  ],
  "pending": [
    {
      "transactionId": "123456789",
      "creditorName": "Claude Renault",
      "creditorAccount": {
        "iban": "NL354543123456900"
      },
      "transactionAmount": {
        "currency": "EUR",
        "amount": "-100.03"
      },
      "valueDate": "2017-10-26",
      "remittanceInformationUnstructured": "Another example
for Remittance Information"
    }
  ],
  "_links": {
    "account": {
      "href": "/v1.1/accounts/qwer3456tzui7890"
    },
    "first": {
```

PSD2 - APIs Implementation Guide v1.1 for TPPs

```
        "href": "/v1.1/accounts/qwer3456tzui7890/transactions?page[number]=1&page[size]=15",
      },
      "previous": {
        "href": "/v1.1/accounts/qwer3456tzui7890/transactions?page[number]=2&page[size]=15"
      },
      "next": {
        "href": "/v1.1/accounts/qwer3456tzui7890/transactions?page[number]=4&page[size]=15"
      },
      "last": {
        "href": "/v1.1/accounts/qwer3456tzui7890/transactions?page[number]=2&page[size]=15"
      }
    }
  }
}
```

Example request to obtain a list of standing orders

GET <https://aspsp.example.es/aspsp-name/v1.1/accounts/qwer3456tzui7890/transactions?bookingStatus=information>

Accept: application/json

X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc

Authorization: Bearer 2YotnFZFEjrlzCsicMWpAA

Consent-ID: 7890-asdf-4321

PSU-IP-Address: 192.168.8.16

PSU-IP-Port: 443

PSU-Accept: application/json

PSU-Accept-Charset: utf-8

PSU-Accept-Encoding: gzip

PSU-Accept-Language: es-ES

PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0) Gecko/20100101 Firefox/54.0

PSD2 - APIs Implementation Guide v1.1 for TPPs

PSU-Http-Method: GET
PSU-Device-ID: f8b3feda-6fe3-11e8-adc0-fa7ae01bbebc
PSU-GEO-Location: GEO:12.526347;54.649862
Date: Sun, 26 Sep 2017 15:02:48 GMT

Example of a standing order list response

HTTP/1.1 200 Ok
X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc
Date: Sun, 26 Sep 2017 15:02:50 GMT
Content-Type: application/json

```
{
  "account": {
    "iban": "ES1111111111111111111111"
  },
  "transactions": {
    "information": [
      {
        "creditorName": "John Miles",
        "creditorAccount": {
          "iban": "ES1111111111111111111111"
        },
        "transactionAmount": {
          "currency": "EUR",
          "amount": "256.67"
        },
        "remittanceInformationUnstructured": "Example for Remittance Information",
        "bankTransactionCode": "PMNT-ICDT-STD0",
        "additionInformationStructured": {
          "standingOrderDetails": {
            "startDate": "2018-03-01",
            "endDate": "2020-06-31",
            "executionRule": "preceding",
            "frequency": "monthly",
            "dayOfExecution": "24"
          }
        }
      }
    ]
  }
}
```

```

    }
  }
]
}
}

```

Example response with error

```

{
  "tppMessages": [{
    "category": "ERROR",
    "code": " ACCESS_EXCEEDED "
  }]
}

```

6.6 AIS: Obtain list of trusted payees

Obtains the list of trusted payees of the PSU, which has given explicit consent.

6.6.1 Request

Endpoint

GET {provider}/{aspsp}/v1.1/trusted-beneficiaries?{account-id}

Path

Field	Description	Type	Mand at.	Format
provider	URL of the ASPSP where the service is published	String	MA	Ex: aspsp.example.es
aspsp	Name of the ASPSP to which the request is to be made.	String	MA	Ex: aspsp-name

Query param

Field	Description	Type	Mand at.	Format
account-id	Reference to the specific PSU account on which to retrieve the list of trusted payees. Only if supported by ASPSP	String	COND	^.{1,100} \$ Ex: aspsp.example.es/v1.1/trusted-beneficiaries?account-id=3dc3d5b3-7023-4848-9853-f5400a64e80f

Header

The same as those defined in the 6.5.1.1

Body

No data travels in the body of this request.

6.6.2 Response

HTTP Code

200 if the request has been successful.

Header

The same as those defined in the section 6.5.1.2

Body

Field	Description	Type	Manda t.	Format
trustedBeneficiaries	This report contains all the trusted beneficiaries of the PSU for those accounts that were consented. This array could be returned empty.	List<TrustedBeneficiary>	MA	Ex: "trustedBeneficiaries": [...]

psuMessage	Text sent to the TPP through the HUB to be displayed to the PSU.	String	OP	^.{1,500} \$ Ex: "psuMessage": "Información para PSU"
tppMessages	Message for the TPP sent through the HUB.	List<Tpp Message>	OP	Ex: "tppMessages": [...]

6.6.3 Examples

Example request to obtain a list of global trust payees

```
GET https://aspsp.example.es/aspsp-name/v1.1/trusted-beneficiaries
Content-Encoding: gzip
Content-Type: application/json
X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc
Authorization: Bearer 2YotnFZFjrlzCsicMWpAA
Consent-ID: 7890-asdf-4321
PSU-IP-Address: 192.168.8.16
PSU-IP-Port: 443
PSU-Accept: application/json
PSU-Accept-Charset: utf-8
PSU-Accept-Encoding: gzip
PSU-Accept-Language: es-ES
PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0
PSU-Http-Method: GET
PSU-Device-ID: f8b3feda-6fe3-11e8-adc0-fa7ae01bbebc
PSU-GEO-Location: GEO:12.526347;54.649862
Date: Sun, 26 Sep 2017 15:02:48 GMT
Date: Sun, 26 Sep 2017 15:02:48 GMT
```

Example of response

Response where the list of trusted payees is global

PSD2 - APIs Implementation Guide v1.1 for TPPs

```
HTTP/1.1 200 Ok
X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc
Date: Sun, 26 Sep 2017 15:02:50 GMT
Content-Type: application/json
{
  "trustedBeneficiaries": [{
    "trustedBeneficiaryId": "8822345-fr33-50df-qqqq",
    "creditorAccount": {
      "iban": "FR761234598765012345"
    },
    "creditorName": "Merchant1Name"
  },
  {
    "trustedBeneficiaryId": "8822345-fr33-50df-qqqq",
    "creditorAccount": {
      "iban": "FR7612345987650126667"
    },
    "creditorName": "Merchant2Name"
  }
]
```

6.7 FCS: Establish consent for funds confirmation service

6.7.1 Fund confirmation consent

With this service a TPP can report a funds confirmation consent to the ASPSP on a specified account.

Unlike the request to establish consent for information about accounts, this consent has no secondary effects on existing ones.

Ex: it does not invalidate a prior consent.

6.7.1.1 Request

Endpoint

POST {provider}/{aspsp}/v2.1/consents/confirmation-of-funds

Path

Field	Description	Type	Mandat.	Format
provider	URL of the HUB where the service is released.	String	MA	Ex: www.hub.com
aspsp	Name of the ASPSP to which the request is to be made.	String	MA	Ex: aspsp-name

Query parameters:

No additional fields are specified.

Header

Field	Description	Type	Mandat.	Format
X-Request-ID	Unique identifier of the operation assigned by the TPP.	String	MA	UUID $^{\wedge}[0-9a-fA-F]\{8\}-[0-9a-fA-F]\{4\}-[0-9a-fA-F]\{4\}-[0-9a-fA-F]\{4\}-[0-9a-fA-F]\{12\}\$$ Ex: X-Request-ID: 1b3ab8e8-0fd5-43d2-946e-d75958b172e7
PSU-ID	Identifier that the PSU uses to identify itself in its ASPSP. It can be reported even if an OAuth token is being used and, in such a case, the ASPSP could check if the PSU-ID and the token match.	String	OP	Ex: PSU-ID: 12345678W
PSU-ID-Type	Type of the PSU-ID. Necessary in scenarios where the PSU has several PSU-IDs as access possibilities.	String	OP	Ex: PSU-ID-Type: NIF

PSD2 - APIs Implementation Guide v1.1 for TPPs

PSU-Corporate-ID	Identifier of "company" in Online Channels.	String	OP	Ex: PSU-Corporate-ID: user@corporate.com
PSU-Corporate-ID-Type	Type of the PSU-Corporate-ID required by the ASPSP to identify its content.	String	OP	Ex: PSU-Corporate-ID-Type: email
Authorization	Bearer Token. Obtained in a previous authentication on OAuth2.	String	MA	Ex: Authorization: Bearer 2YotnFZFEjr1zCsicMWpAA
TPP-Redirect-Preferred	<p>If "true", the TPP has communicated to the HUB that it prefers SCA by redirection.</p> <p>If "false", the TPP has informed the HUB that it prefers not to be redirected to SCA and the procedure will be by decoupled flow.</p> <p>If the parameter is not used, the ASPSP will choose the SCA flow to apply depending on the SCA method chosen by the TPP / PSU.</p> <p>EMBEDDED NOT SUPPORTED IN THIS VERSION</p>	Boolean	OP	Ex: TPP-Redirect-Preferred: true
TPP-Redirect-URI	<p>URI of the TPP where the transaction flow must be redirected after any of the SCA phases.</p> <p>It is recommended to always use this header field.</p> <p>In the future, this field could change to mandatory.</p>	String	COND	<p>^.{1,250}\$</p> <p>Ex: TPP-Redirect-URI: "https://tpp.example.es/cb"</p>

	The domain of this URI is required to be the same as the content in the TPP web certificate.			
TPP-Nok-Redirect-URI	<p>If this URI is contained, the TPP is requesting to redirect the transaction flow to this address instead of the TPP-Redirect-URI in case of a negative result of the SCA method by redirection.</p> <p>The domain of this URI is required to be the same as the content in the TPP web certificate.</p>	String	OP	<p>^.{12,50}\$</p> <p>Ex: TPP-Nok-Redirect-URI:"https://tpp.example.es/cb/nok"</p>
TPP-Explicit-Authorisation-Preferred	<p>If equal to true, the TPP chooses to initiate the authorisation process separately, e.g. due to the need for authorisation of a set of operations simultaneously.</p> <p>If false or the parameter is not used, there is no TPP preference. The TPP takes a direct authorisation of the transaction in the next step.</p> <p>Note: ASPSP might not take it into account if it doesn't support it.</p>	Boolean	OP	<p>Ex: TPP-Explicit-Authorisation-Preferred: false</p>

TPP-Brand-Logging-Information	This field could be used by the TPP to inform the ASPSP about the brand used by the TPP for the PSU. This information can be used to improve communication between the ASPSP and the PSU or the ASPSP and the TPP.	String	OP	$\wedge.\{1,70\}\$$ Ex: TPP-Brand-Logging-Information: TPP Brand
--------------------------------------	--	--------	----	---

Body

Field	Description	Type	Mandat.	Format
account	Account on which the fund consultation is to be carried out.	Account Reference	MA	Ex: "access": {...}
cardExpiryDate	Expiry date of the card issued by PIISP.	String	OP	ISODate Ex: "validUntil": "2018-05-17"
cardInformation	Additional explanation of the product.	String	OP	$\wedge.\{1,140\}\$$
registrationInformation	Additional information about the registration process for the PSU. For example, a reference to the TPP/PSU contract.	String	OP	$\wedge.\{1,140\}\$$

6.7.1.2 Response

HTTP Code

201 if the resource has been created

Response code

HTTP 201 response code if resource is created successfully.

Header

PSD2 - APIs Implementation Guide v1.1 for TPPs

Field	Description	Type	Mandat.	Format
Location	Contains the hyperlink to the generated resource	String	MA	Max512Text Ex: Location: /v2.1/consents/confirmation-of-funds/{consentId}
X-Request-ID	Unique identifier of the operation assigned by the TPP.	String	MA	UUID ^[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}\$ Ex: X-Request-ID: 1b3ab8e8-0fd5-43d2-946e-d75958b172e7
ASPSP-SCA-Approach	Value returned if the SCA method has been set. Possible values: <ul style="list-style-type: none">REDIRECT The OAuth based SCA will be taken as REDIRECT.	String	COND	Ex: ASPSP-SCA-Approach: REDIRECT

Body

Field	Description	Type	Mandat.	Format
consentStatus	Consent State Defined values in 9.5 Consent states	String	MA	Ex: "consentStatus": "received"
consentId	Identifier of the resource that refers to the consent. It must be contained if consent was generated.	String	MA	^.{1,36}\$ Ex: "consentId": "123-QWE-456"

scaMethods	<p>This element is contained if SCA is required and if the PSU can choose between different authentication methods.</p> <p>If this data is contained, the link "startAuthorisationWithAuthenticationMethodSelection" will also be reported.</p> <p>These methods must be presented to the PSU.</p> <p>Note: Only if ASPSP supports SCA method selection</p>	List<AuthenticationObject>	COND	Ex: "scaMethods": [...]
_links	<p>List of hyperlinks to be recognized by the TPP. Supported types in this response:</p> <ul style="list-style-type: none"> scaRedirect: in case of SCA by redirection. Link where the PSU browser must be redirected by the TPP. startAuthorisation: in case an explicit start of transaction authorization is required (no SCA method selection) startAuthorisationWithAuthenticationMethodSelection: link to the authorisation end-point where the authorisation sub-resource has to be generated while selecting the SCA method. This link is 	Links	MA	Ex: "_links": {...}

	<p>contained under the same conditions as the "scaMethods" field</p> <ul style="list-style-type: none"> • self: link to the resource created by this request. • state: link to retrieve the state of the transaction. • scaStatus: link to consult the SCA state corresponding to the authorisation sub-resource. This link is only contained if an authorization sub-resource has been created. 			
psuMessage	Text to show to the PSU.	String	OP	$\wedge.\{1,500\}$ \$ Ex: "psuMessage": "Información para PSU"
tppMessages	Message to the TPP	List<TppMessage>	OP	Ex: "tppMessages": [...]

6.7.1.3 Examples

Example of consent request

POST <https://www.hub.com/aspsp-name/v2.1/consent/confirmation-of-funds>

Content-Encoding: gzip

Content-Type: application/json

X-Request-ID: 10391c7e-ad88-49ec-a2ad-00aacb1f6541

Authorization: Bearer 2YotnFZFjrlzCsicMWpAA

PSU-IP-Address: 192.168.8.16

PSU-IP-Port: 443

PSU-Accept: application/json

PSD2 - APIs Implementation Guide v1.1 for TPPs

```
PSU-Accept-Charset: utf-8
PSU-Accept-Encoding: gzip
PSU-Accept-Language: es-ES
PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0
PSU-Http-Method: POST
PSU-Device-ID: f8b3feda-6fe3-11e8-adc0-fa7ae01bbebc
PSU-GEO-Location: GEO:12.526347;54.649862
TPP-Redirect-Preferred: true
TPP-Redirect-URI: https://www.tpp.com/cb
TPP-Nok-Redirect-URI: https://www.tpp.com/cb/nok
Date: Sun, 26 Sep 2017 15:02:37 GMT
{
  "account": {
    "iban": "ES1111111111111111111111"
  },
  "cardExpiryDate": "2020-12-31",
  "cardInformation": "MyMerchant Loyalty Card",
  "registrationInformation": "Your contrat Number 1234 with
MyMerchant is completed with the registration with your bank."
}
```

Example response in case of SCA by redirection with implicitly generated authorisation sub-resource

```
HTTP/1.1 201 Created
X-Request-ID: 10391c7e-ad88-49ec-a2ad-00aacb1f6541
ASPSP-SCA-Approach: REDIRECT
Date: Sun, 26 Sep 2017 15:02:43 GMT
Location: /v2.1/consents/confirmation-of-funds/123-asdf-456
Content-Type: application/json
{
  "consentStatus": "received",
  "consentId": "123-asdf-456",
  "_links": {
    "scaRedirect": {
      "href": "https://hub.example.es/authorization "
    },
    "self": {
```

```

        "href":      "/v2.1/consents/confirmation-of-funds/123-
asdf-456",
    },
    "state": {
        "href":      "/v2.1/consents/confirmation-of-funds/123-
asdf-456",
    },
    "scaStatus": {
        "href":      "/v2.1/consents/123-asdf-
456/authorisations/confirmation-of-funds/123auth456"
    }
}
}

```

6.7.2 Obtain consent state

This service allows the TPP to know the state of a previously initiated consent request.

6.7.2.1 Request

Endpoint

GET {provider}/{aspsp}/v2.1/consents/confirmation-of-funds/{consent-id}/state

Path

Field	Description	Type	Manda t.	Format
provider	URL of the HUB where the service is released.	String	MA	Ex: www.hub.com
aspsp	Name of the ASPSP to which the request is to be made.	String	MA	Ex: aspsp-name
consentId	Identifier of the resource that refers to the consent.	String	MA	^.{1,36}\$ Ex:123-qwerty-456

PSD2 - APIs Implementation Guide v1.1 for TPPs

	Previously sent in response to a consent request message from the TPP.			
--	--	--	--	--

Query parameters:

No additional fields are specified.

Header

Field	Description	Type	Mandate	Format
X-Request-ID	Unique identifier of the request assigned by the TPP.	String	MA	UUID ^[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}\$ Ex: X-Request-ID: 1b3ab8e8-0fd5-43d2-946e-d75958b172e7
Authorization	Bearer Token. Obtained in a previous authentication on OAuth2.	String	MA	Ex: Authorization: Bearer 2YotnFZFEjr1zCsicMWpAA

Body

No additional data travels.

6.7.2.2 Response

This message is returned to the TPP in response to the consent state request message.

Response code

HTTP Code

200 if the request has been successful.

Header

Field	Description	Type	Mandat.	Format
X-Request-ID	Unique identifier of the request assigned by the TPP.	String	MA	UUID $^{\wedge}[0-9a-fA-F]\{8\}-[0-9a-fA-F]\{4\}-[0-9a-fA-F]\{4\}-[0-9a-fA-F]\{4\}-[0-9a-fA-F]\{12\}\$$ Ex: X-Request-ID: 1b3ab8e8-0fd5-43d2-946e-d75958b172e7

Body

Field	Description	Type	Mandat.	Format
consentStatus	Consent authentication state. Defined values in 9.5 Consent states	String	MA	Ex: "consentStatus": "valid"
psuMessage	Text to show to the PSU	String	OP	$^{\wedge}.\{1,500\}\$$ Ex: "psuMessage": "Información para PSU"
tppMessages	Message to the TPP	List<Tp pMessage>	OP	Ex: "tppMessages": [...]

6.7.2.3 Examples

Example of request

GET <https://www.hub.com/aspsp-name/v2.1/consents/confirmation-of-funds/123asdf456/state>

Accept: application/json

X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc

Authorization: Bearer 2YotnFZFEjrlzCsicMWpAA

PSD2 - APIs Implementation Guide v1.1 for TPPs

```
PSU-IP-Address: 192.168.8.16
PSU-IP-Port: 443
PSU-Accept: application/json
PSU-Accept-Charset: utf-8
PSU-Accept-Encoding: gzip
PSU-Accept-Language: es-ES
PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0
PSU-Http-Method: GET
PSU-Device-ID: f8b3feda-6fe3-11e8-adc0-fa7ae01bbebc
PSU-GEO-Location: GEO:12.526347;54.649862
Date: Sun, 26 Sep 2017 15:02:48 GMT
```

Example response

```
HTTP/1.1 200 Ok
X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc
Date: Sun, 26 Sep 2017 15:02:50 GMT
Content-Type: application/json
{
  "consentStatus": "valid"
}
```

6.7.3 Retrieve consent information

6.7.3.1 Request

This message is sent by the TPP as a request to retrieve information from a previously created fund confirmation consent. Especially useful for the TPP in cases where consent was managed directly between the ASPSP and the PSU.

Endpoint

GET {provider}/{aspsp}/v2.1/consents/confirmation-of-funds/{consentId}

Path

Field	Description	Type	Mandant.	Format

PSD2 - APIs Implementation Guide v1.1 for TPPs

provider	URL of the HUB where the service is released.	String	MA	Ex: www.hub.com
aspsp	Name of the ASPSP to which the request is to be made.	String	MA	Ex: aspsp-name
consentId	Identifier of the resource that refers to the consent. Previously sent in response to a consent request message from the TPP.	String	MA	^. {1,36} \$ Ex: 7890-asdf-4321

Query parameters:

No additional fields are specified.

Header

The same as those defined in the section 6.7.2.1

Body

No additional data travels.

6.7.3.2 Response

This message is returned to the TPP in response to the message to retrieve consent information.

HTTP Code

200 if the request has been successful.

Header

The same as those defined in the section 6.7.2.2

Body

Field	Description	Type	Mand at.	Format
-------	-------------	------	----------	--------

PSD2 - APIs Implementation Guide v1.1 for TPPs

account	Account on which the fund consultation is to be carried out.	Account Reference	MA	Ex: "access": {...}
cardExpiryDate	Expiry date of the card issued by PIISP.	String	OP	ISODate Ex: "validUntil": "2018-05-17"
cardInformation	Additional explanation of the product.	String	OP	^. {1,140}\$
registrationInformation	Additional information about the registration process for the PSU. For example, a reference to the TPP/PSU contract.	String	OP	^. {1,140}\$
consentStatus	Consent State Values defined in annexes.	String	MA	Ex: "consentStatus": "valid"
psuMessage	Text sent to the TPP through the HUB to be displayed to the PSU.	String	OP	^. {1,500} \$ Ex: "psuMessage": "Información para PSU"
tppMessages	Message to the TPP	List<Tp pMessage>	OP	Ex: "tppMessages": [...]

6.7.3.3 Examples

Example of request

GET <https://www.hub.com/aspsp-name/v2.1/consents/confirmation-of-funds/7890-asdf-4321/>

Accept: application/json

X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc

Authorization: Bearer 2YotnFZFEjrlzCsicMWpAA

PSU-IP-Address: 192.168.8.16

PSU-IP-Port: 443

PSU-Accept: application/json

PSU-Accept-Charset: utf-8

PSU-Accept-Encoding: gzip

PSU-Accept-Language: es-ES

PSD2 - APIs Implementation Guide v1.1 for TPPs

```
PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0
PSU-Http-Method: GET
PSU-Device-ID: f8b3feda-6fe3-11e8-adc0-fa7ae01bbebc
PSU-GEO-Location: GEO:12.526347;54.649862
Date: Sun, 26 Sep 2017 15:02:48 GMT
```

Example response

```
HTTP/1.1 200 Ok
X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc
Date: Sun, 26 Sep 2017 15:02:50 GMT
Content-Type: application/json
{
  "account": {
    "iban": "ES1111111111111111111111"
  },
  "cardExpiryDate": "2020-12-31",
  "cardInformation": "MyMerchant Loyalty Card",
  "registrationInformation": "Your contrat Number 1234 with
MyMerchant is completed with the registration with your bank."
  "consentStatus": "valid"
}
```

6.7.4 Revoke consent

6.7.4.1 Request

This service allows you to request the deletion of a consent previously created in the ASPSP.

Endpoint

DELETE {provider}/{aspsp}/v2.1/consents/confirmation-of-funds/{consentId}

Path

Field	Description	Type	Mandat.	Format
-------	-------------	------	---------	--------

PSD2 - APIs Implementation Guide v1.1 for TPPs

provider	URL of the HUB where the service is released.	String	MA	Ex: www.hub.com
aspsp	Name of the ASPSP to which the request is to be made.	String	MA	Ex: aspsp-name
consentId	Identifier of the resource that refers to the consent. Previously sent in response to a consent request message from the TPP.	String	MA	^. {1,36} \$ Ex: 7890-asdf-4321

Query parameters:

No additional fields are specified.

Header

The same as those defined in the section 6.7.2.1

Body

No additional data travels.

6.7.4.2 Response

This message is sent to the TPP in response to the request to remove consent.

Response code

HTTP 204 response code for successful cancellation.

Header

The same as those defined in the section 6.7.2.2

Body

No additional fields are specified.

6.7.4.3 Examples

Example of request

```
DELETE https://www.hub.com/aspsp-name/v2.1/consents/confirmation-of-funds/7890-asdf-4321
Accept: application/json
X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc
Authorization: Bearer 2YotnFZFEjrlzCsicMWpAA
PSU-IP-Address: 192.168.8.16
PSU-IP-Port: 443
PSU-Accept-Charset: utf-8
PSU-Accept-Encoding: gzip
PSU-Accept-Language: es-ES
PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0
PSU-Http-Method: DELETE
PSU-Device-ID: f8b3feda-6fe3-11e8-adc0-fa7ae01bbebc
PSU-GEO-Location: GEO:12.526347;54.649862
Date: Sun, 26 Sep 2017 15:02:48 GMT
```

Example response

```
HTTP / 1.1 204 Ok
X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc
Date: Sun, 26 Sep 2017 15:02:50 GMT
```

6.8 FCS: Fund Confirmation Service

6.8.1 Fund inquiry

This type of message is used in the fund enquiry service. The TPP sends the HUB the request for a fund inquiry for a given amount.

The HUB contacts the ASPSP to ask whether or not it has funds and, after consultation, returns the response to the TPP.

6.8.1.1 Request

Endpoint

POST {provider}/{aspsp}/v1.1/funds-confirmations

Path

Field	Description	Type	Mandat.	Format
provider	URL of the HUB where the service is released	String	MA	Ex: www.hub.com
aspsp	Name of the ASPSP to which the request is to be made.	String	MA	Ex: aspsp-name

Header

Field	Description	Type	Mandat.	Format
X-Request-ID	Unique identifier of the operation assigned by the TPP.	String	MA	UUID ^[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}\$ Ex: X-Request-ID: 1b3ab8e8-0fd5-43d2-946e-d75958b172e7
Authorization	Bearer Token. Obtained in a previous authentication on OAuth2. Only if consent management has been carried out through the API.	String	MA	Ex: Authorization: Bearer 2YotnFZFEjr1zCsicMWpAA

Consent-ID	Identifier of the consent obtained in the transaction to request consent. Only if consent management has been carried out through the API.	String	MA	^.{1,36}\$ Ex: Consent-ID: 7890-asdf-4321
-------------------	---	--------	----	--

Body

Field	Description	Type	Mandat.	Format
account	PSU account number.	AccountReference	MA	Ex: "account": {"iban": "ES1111111111111111"}
instructedAmount	Contains the amount and currency to consult.	Amount	MA	Ex: "instructedAmount": {...}

6.8.1.2 Response

This message is returned by the HUB to the TPP in response to the funds confirmation message.

HTTP Code

200 if the request has been successful.

Header

Field	Description	Type	Mandat.	Format
X-Request-ID	Unique identifier of the operation assigned by the TPP and sent through the HUB to the ASPSP.	String	MA	UUID

				[^] [0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}\$ Ex: X-Request-ID: 1b3ab8e8-0fd5-43d2-946e-d75958b172e7
--	--	--	--	--

Body

Field	Description	Type	Mandate	Format
fundsAvailable	It takes the value "true" if there are sufficient funds available at the time of the request; "false" otherwise.	Boolean	MA	Ex: "fundsAvailable": true
tppMessages	Message to the TPP	List<Tpp Message>	OP	Ex: "tppMessages": [...]

6.8.1.3 Examples

Example of request

POST <https://www.hub.com/aspsp-name/v1.1/funds-confirmations>

Content-Encoding: gzip

Content-Type: application/json

X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc

Authorization: Bearer 2YotnFZFEjrlzCsicMWpAA

Consent-ID: 7890-asdf-4321

Date: Sun, 17 Oct 2017 13:15:17 GMT

```
{
  "account": {
    "iban": "ES1111111111111111111111"
  },
  "payee": "Name123",
  "instructedAmount": {
```

```
        "currency": "EUR",  
        "amount": "153.50"  
    }  
}
```

Sample response with available funds

```
HTTP/1.1 200 Ok  
X-Request-ID: 0ee25bf4-6ff1-11e8-adc0-fa7ae01bbebc  
Date: Sun, 26 Sep 2017 15:02:47 GMT  
Content-Type: application/json  
{  
    "fundsAvailable": true  
}
```

6.9 Sessions: combination of AIS and PIS services

Session support allows combining AIS and PIS services in the same session.

The session support is determined by the access token obtained after performing the OAuth2 protocol (pre-step)

For the session to be supported, the access token must have been obtained for the "PIS" and "AIS" scope and, the TPP, have the PISP and AISP roles in its eIDAS certificate.

6.10 Processes common to services

6.10.1 Start the authorization process (explicit)

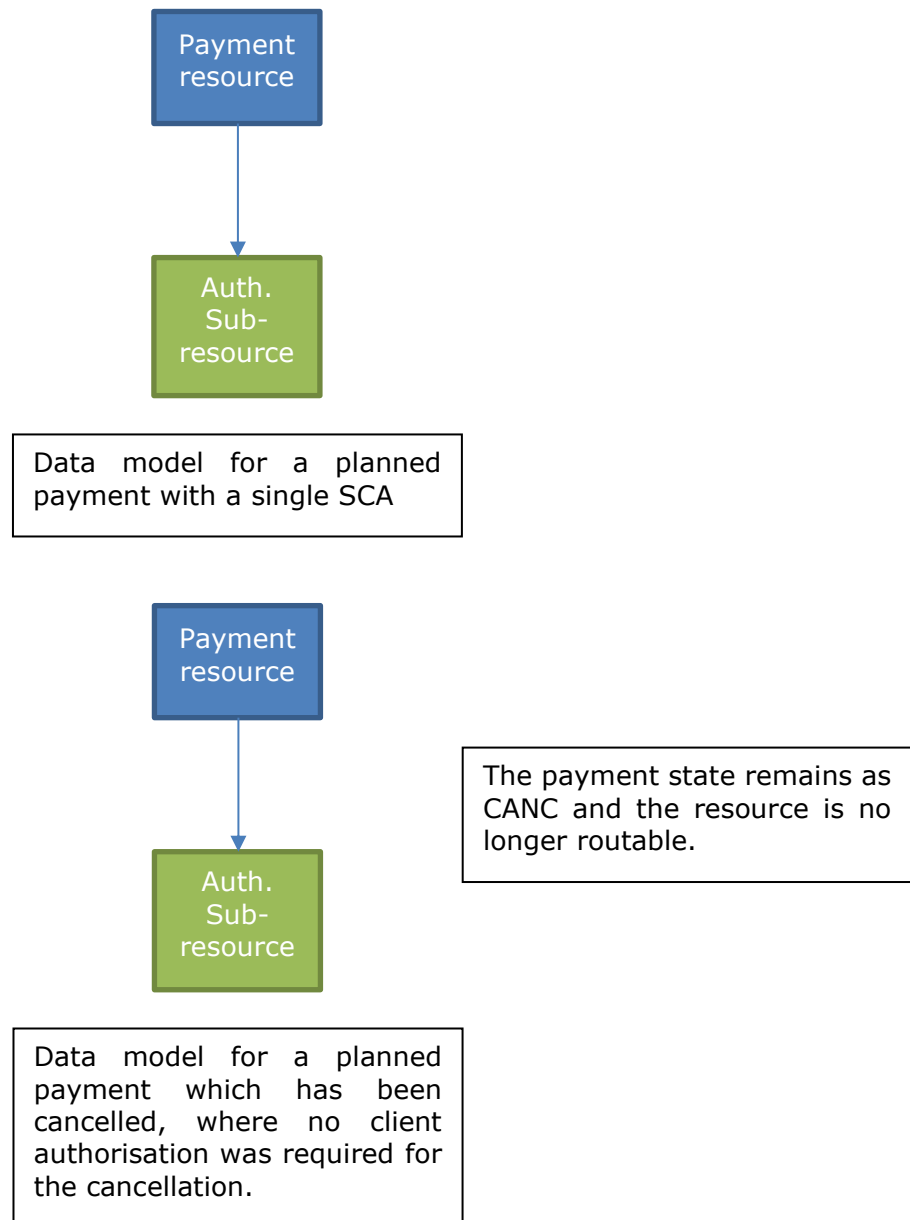
Use

The initiate authorization process is a process required to create a new authorization sub-resource (if not created implicitly). Applies in the following scenarios:

- The ASPSP has indicated with a "startAuthorisation" link in the response to a payment initiation request that an explicit start of the authorisation process is required by the TPP.
- The ASPSP has indicated with a "startAuthorisation" link in the response to a request for consent to account information that an explicit start of the authorisation process is required by the TPP.

- The ASPSP has indicated with a "startAuthorisation" link in the response to a Confirmation of Funds consent request that an explicit start of the authorisation process is required by the TPP.

Abstract data model



6.10.1.1 Request

Endpoint in case of Fund Confirmation Consent

POST {provider}/{aspsp}/v2.1/consents/confirmation-of-funds/{consentId}/authorisations

Endpoint in case of Start of Payment

POST {provider}/{aspsp}/v1.1/{payment-service}/{payment-product}/{paymentId}/authorisations

Endpoint in case of Account Information Consent

POST {provider}/{aspsp}/v1.1/consents/{consentId}/authorisations

Path

Field	Description	Type	Mandat.	Format
provider	URL of the HUB where the service is released.	String	MA	Ex: hub.example.es
aspsp	Name of the ASPSP to which the request is to be made.	String	MA	Ex: aspsp-name
payment-service	Possible values are: <ul style="list-style-type: none"> payments periodic-payments 	String	COND	Ex: {provider}/v1.1/payments
payment-product	Paid product to use. List of supported products: <ul style="list-style-type: none"> sepa-credit-transfers instant-sepa-credit-transfers target-2-payments cross-border-credit-transfers 	String	COND	Ex: {provider}/v1.1/payments/sepa-credit-transfers/
paymentId, consentId	Resource identifier that refers to the initiation of payment.	String	MA	^.{1,36}\$ Ex: 123-qwe-456

Query parameters:

No additional parameters are specified for this request.

Header

Field	Description	Type	Mand at.	Format
Content-Type	Value: application / json	String	MA	Content-Type: application/json
X-Request-ID	Unique transaction identifier assigned by the TPP and forwarded via the HUB to the ASPSP	String	MA	UUID ^[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}\$ Ex: X-Request-ID: 1b3ab8e8-0fd5-43d2-946e-d75958b172e7
Authorization	Bearer Token. Obtained in a previous authentication on OAuth2.	String	MA	Ex: Authorization: Bearer 2YotnFZFEjr1zCsicMWpAA
PSU-ID	Identifier that the PSU uses to identify itself in its ASPSP. It can be reported even if an OAuth token is being used and, in such a case, the ASPSP could check if the PSU-ID and the token match.	String	OP	Ex: PSU-ID: 12345678W
PSU-ID-Type	Type of the PSU-ID. Necessary in scenarios where the PSU has several PSU-IDs as access possibilities.	String	OP	Ex: PSU-ID-Type: NIF
PSU-Corporate-ID	Identifier of "company" in Online Channels.	String	OP	Ex: PSU-Corporate-ID: user@corporate.com
PSU-Corporate-ID-Type	Type of the PSU-Corporate-ID required by the ASPSP to identify its content. TBD	String	OP	Ex: PSU-Corporate-ID-Type: email

TPP-Redirect-Preferred	<p>If "true", the TPP has communicated to the HUB that it prefers SCA by redirection.</p> <p>If "false", the TPP has informed the HUB that it prefers not to be redirected to SCA and the procedure will be by decoupled flow.</p> <p>If the parameter is not used, the ASPSP will choose the SCA flow to apply depending on the SCA method chosen by the TPP / PSU.</p> <p>EMBEDDED NOT SUPPORTED IN THIS VERSION</p>	Boolean	OP	Ex: TPP-Redirect-Preferred: true
TPP-Redirect-URI	<p>HUB URI where the flow of the transaction should be redirected after finishing the SCA by redirect.</p> <p>It is recommended to always use this header field.</p> <p>In the future, this field could change to mandatory.</p>	String	COND	<p>^.{1,250}\$</p> <p>Ex: TPP-Redirect-URI:"https://hub.example.es/cb"</p>
TPP-Nok-Redirect-URI	<p>If this URI is contained, the TPP is requesting to redirect the transaction flow to this address instead of the TPP-Redirect-URI in case of a negative result of the SCA method by redirection.</p>	String	OP	<p>^.{1,250}\$</p> <p>Ex: TPP-Nok-Redirect-URI:"https://hub.example.es/cb/nok"</p>

Body

No additional fields are specified.

6.10.1.2 Response

HTTP Code

201 if the resource has been created

Header

Field	Description	Type	Mand at.	Format
Location	Contains the link to the generated resource.	String	MA	Ex: Location: /v1.1/payments/{pay ment- product}/{paymentId} /authorisations/123qw ert/456
X-Request-ID	Unique identifier of the operation assigned by the TPP and sent through the HUB to the ASPSP.	String	MA	UUID ^[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}\$ Ex: X-Request-ID: 1b3ab8e8-0fd5-43d2-946e-d75958b172e7
ASPSP-SCA-Approach	Value returned if the SCA method has been set. Possible values: <ul style="list-style-type: none">• REDIRECT The SCA based on OAuth2 will be taken as REDIRECT.	String	COND	Ex: ASPSP-SCA-Approach: REDIRECT

Body

Field	Description	Type	Mand at.	Format
scaStatus	SCA state	String	MA	Ex: "scaStatus": "received"

authorizationId	Resource identifier that refers to the authorization sub-resource created.	String	MA	^.{1,36}\$ Ex: "authorisationId": "1b3ab8e8-0fd5-43d2-946e-d75958b172e7"
_links	List of hyperlinks to be recognized by the TPP. Supported types in this response: <ul style="list-style-type: none"> scaRedirect: in case of SCA by redirection. Link where the PSU browser must be redirected by the TPP. scaStatus: link to consult the SCA state corresponding to the authorisation sub-resource. 	Links	MA	Ex: "_links": {...}
psuMessage	Text sent to the TPP through the HUB to be displayed to the PSU.	String	OP	^.{1,512}\$ Ex: "psuMessage": "Información para PSU"
tppMessages	Message for the TPP sent through the HUB.	List<Tp pMessage>	OP	Ex: "tppMessages": [...]

6.10.2 Get authorization sub-resources

It will provide an array of resource identifiers for all generated authorization sub-resources.

6.10.2.1 Request

Endpoint in case of Start of Payment

PSD2 - APIs Implementation Guide v1.1 for TPPs

GET {provider}/{aspsp}/v1.1/{payment-service}/{payment-product}/{paymentId}/authorisations

Endpoint in case of Account Information Consent

GET {provider}/{aspsp}/v1.1/consents/{consentId}/authorisations

Path

Field	Description	Type	Mandant.	Format
provider	URL of the ASPSP where the service is published.	String	MA	Ex: hub.example.es
aspsp	Name of the ASPSP to which the request is to be made.	String	MA	Ex: aspsp-name
payment-service	Possible values are: <ul style="list-style-type: none">• payments• periodic-payments	String	COND	Ex: {provider}/v1.1/payments
payment-product	Paid product to use. List of supported products: <ul style="list-style-type: none">• sepa-credit-transfers• instant-sepa-credit-transfers• target-2-payments• cross-border-credit-transfers	String	COND	Ex: {provider}/v1.1/payments/sepa-credit-transfers/
paymentId, consentId	Resource identifier that refers to the initiation of payment.	String	MA	^. {1,36}\$ Ex: 123-qwe-456

Query parameters:

No additional fields are specified.

Header

The same as those defined in the section **Error! No se encuentra el origen de la referencia.**

Body

No additional data is specified.

6.10.2.2 Response

HTTP Code

200 if the request has been successful.

Header

The same as those defined in the section **Error! No se encuentra el origen de la referencia.**

Body

Field	Description	Type	Mand at.	Format
authorisat ionIds	Array of authorizationIds. Note: required field if it is not a cancellation	Array<S tring>	COND	^.{1,36}\$ Ex: " authorisationIds ": [...]
psuMessa ge	Text sent to the TPP through the HUB to be displayed to the PSU.	String	OP	^.{1,500} \$ Ex: "psuMessage": "Información para PSU"
tppMessag es	Message for the TPP sent through the HUB.	List<Tp pMessa ge>	OP	Ex: "tppMessages": [...]

6.10.2.3 Examples

Example of request

GET <https://hub.example.es/asp-name/v1.1/payments/sepa-credit-transfers/123-qwe-456/cancellation-authorisations>

X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc

Authorization: Bearer 2YotnFZFjrlzCsicMWpAA

PSU-IP-Address: 192.168.8.16

Content-Type: application/json

Date: Sun, 26 Sep 2017 15:02:48 GMT

Example response

```
HTTP/1.1 200 Ok
X-Request-ID: 0ee25bf4-6ff1-11e8-adc0-fa7ae01bbebc
Date: Sun, 26 Sep 2017 15:02:47 GMT
{
  "authorizationIds": ["123auth456"]
}
```

6.10.3 Get SCA state

Message sent by the TPP to the ASPSP through the Hub to request the SCA state of an authorization sub-resource.

6.10.3.1 Request

Endpoint in case of Start of Payment

GET {provider}/{aspsp}/v1.1/{payment-service}/{payment-product}/{paymentId}/authorisations/{authorisationId}

Endpoint in case of Account Information Consent

GET
{provider}/{aspsp}/v1.1/consents/{consentId}/authorisations/{authorisationId}

Endpoint in case of Fund Confirmation Consent

GET {provider}/{aspsp}/v2.1/consents/confirmation-of-funds/{consentId}/authorisations/{authorisationId}

Path

Field	Description	Type	Mandat.	Format
provider	URL of the HUB where the service is released.	String	MA	Ex: hub.example.es
aspsp	Name of the ASPSP to which the request is to be made.	String	MA	Ex: aspsp-name

payment-service	Possible values are: <ul style="list-style-type: none"> • payments • periodic-payments 	String	COND	Ex: {provider}/v1.1/ payments
payment-product	Paid product to use. List of supported products: <ul style="list-style-type: none"> • sepa-credit-transfers • instant-sepa-credit-transfers • target-2-payments • cross-border-credit-transfers 	String	COND	Ex: {provider}/v1.1/ payments/sepa-credit-transfers/
paymentId, consentId	Resource identifier referring to the initiation of payment or consent	String	MA	^. {1,36}\$ Ex: 123-qwe-456
authorizationId	Identifier of the sub-resource associated with the consent.	String	COND	^. {1,36}\$

Query parameters:

No additional fields are specified.

Header

The same as those defined in the section **Error! No se encuentra el origen de la referencia.**

Body

No additional data is specified.

6.10.3.2 Response

HTTP Code

200 if the request has been successful.

Header

The same as those defined in the section **Error! No se encuentra el origen de la referencia.**

Body

Field	Description	Type	Manda t.	Format
scaStatus	SCA state	String	MA	Ex: "scaStatus": "finalised"
trustedBeneficiaryFlag	With this flag the ASPSP could optionally communicate to the TPP that the creditor was part of the list of trusted payees. This attribute is only contained in case of a final state of the scaStatus.	Boolean	OP	Ex: `trustedBeneficiaryFlag`: true
psuMessage	Text sent to the TPP through the HUB to be displayed to the PSU.	String	OP	^.{1,500} \$ Ex: "psuMessage": "Información para PSU"
tppMessages	Message for the TPP sent through the HUB.	List<TppMessage>	OP	Ex: "tppMessages": [...]

6.10.3.3 Examples

Example of request

GET <https://hub.example.es/aspsp-name/v1.1/payments/sepa-credit-transfers/123-qwe-456/cancellation-authorisations/123asd456>

X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc

Authorization: Bearer 2YotnFZFjrlzCsicMWpAA

PSU-IP-Address: 192.168.8.16

PSU-IP-Port: 443

PSU-Accept: application/json

PSU-Accept-Charset: utf-8

PSU-Accept-Encoding: gzip

PSU-Accept-Language: es-ES

PSD2 - APIs Implementation Guide v1.1 for TPPs

```
PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0
PSU-Http-Method: GET
PSU-Device-ID: f8b3feda-6fe3-11e8-adc0-fa7ae01bbebc
PSU-GEO-Location: GEO:12.526347;54.649862
Date: Sun, 26 Sep 2017 15:02:48 GMT
```

Example response

```
HTTP/1.1 200 Ok
X-Request-ID: 96201400-6ff9-11e8-adc0-fa7ae01bbebc
Date: Sun, 26 Sep 2017 15:02:50 GMT
Content-Type: application/json
{
    "scaStatus": "finalised"
}
```

7. DESCRIPTION SERVICES OF ADDED VALUE

7.1 ASPSPs service available

This message is sent by the TPP to the HUB to receive information about which ASPSPs are available in the system.

7.1.1 Version 1

7.1.1.1 Request

Endpoint

GET {provider}/v1.1/sva/aspsps

Path

Field	Description	Type	Mandatory	Format
provider	URL of the HUB where the service is released.	String	MA	Ex: www.hub.com

Header

Field	Description	Type	Mandatory	Format
X-Request-ID	Unique identifier of the operation assigned by the TPP.	String	MA	UUID ^[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}\$ Ex: X-Request-ID: 1b3ab8e8-0fd5-43d2-946e-d75958b172e7

Body

No additional fields are specified.

7.1.1.2 Response

HTTP Code

200 if the request has been successful.

Field	Description	Type	Mandat.	Format
aspsps	List of ASPSPs available in the system. The returned list will be made up of relevant information from the ASPSP.	List<Aspsp>	MA	Ex: "aspsps": []
tppMessages	Contains the type of message and the code associated with it	TppMessage	MA	Ex: "tppMessages": [...]

7.1.1.3 Examples

Example of request

```
GET https://www.hub.com/v1.1/sva/aspsps
Content-Encoding: gzip
Content-Type: application/json
X-Request-ID: 29391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date: Sun, 27 Oct 2017 13:15:17 GMT
```

Example response

```
HTTP/1.1 200 Ok

{
  "aspsps": [
    {
      "bic": "XXXXXESMMXXX",
      "name": "aspsp1"
    },
    {
      "bic": "YYYYXESMMXXX",
```

```

        "name": "aspsp2"
      }
    ]
  }

```

7.1.2 Version 2

This version includes the API name for each ASPSP.

7.1.2.1 Request

Endpoint

GET {provider}/v2.1/sva/aspsps

Path

Field	Description	Type	Manda t.	Format
provider	URL of the HUB where the service is released.	String	MA	Ex: www.hub.com

Header

Field	Description	Type	Manda t.	Format
X-Request-ID	Unique identifier of the operation assigned by the TPP.	String	MA	UUID ^[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}\$ Ex: X-Request-ID: 1b3ab8e8-0fd5-43d2-946e-d75958b172e7

Body

No additional fields are specified.

7.1.2.2 Response

Field	Description	Type	Mandat.	Format
aspsps	List of ASPSPs available in the system. The returned list will be made up of relevant information from the ASPSP.	List<Aspsp>	MA	Ex: "aspsps": []
tppMessages	Contains the type of message and the code associated with it	TppMessage	MA	Ex: "tppMessages": [...]

7.1.2.3 Examples

Example of request

```
GET https://www.hub.com/v2.1/sva/aspsps
Content-Encoding: gzip
Content-Type: application/json
X-Request-ID: 29391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date: Sun, 27 Oct 2017 13:15:17 GMT
```

Example response

```
HTTP/1.1 200 Ok

{
  "aspsps": [
    {
      "bic": "XXXXESMMXXX",
      "name": "Nombre del banco",
      "apiName": "nombreBanco1"
    },
    {
      "bic": "YYYYESMMXXX",
      "name": "Nombre del banco 2",
      "apiName": "nombreBanco2"
    }
  ]
}
```

```
        }  
    ]  
}
```

7.2 SVA: start of payment with list of accounts available for PISP

This service allows the TPP to initiate a payment without informing the issuer's account "debtorAccount" and provides the list of accounts during the SCA flow for the PSU to select one.

This valuable service complements the payments API and makes use of CORE services to:

- Get payment state
- Retrieve payment initiation information
- Cancel start of payment

7.2.1 Payment Initiation Flows

7.2.1.1 SCA flow by redirection with account selection: implicit start of authorization process

The following represents the sequence of requests / responses that are necessary for this service.

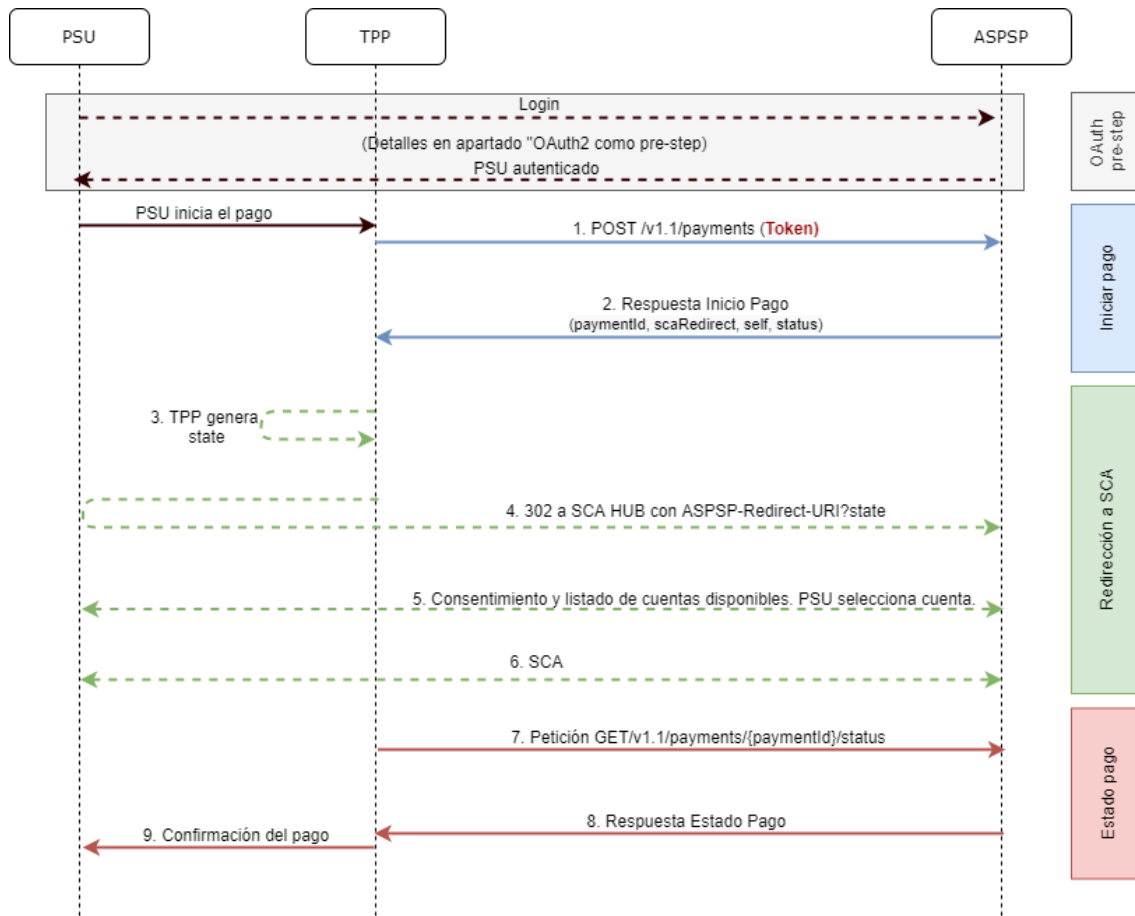


Figure 6: SCA flow by redirection with account selection- implicit start of authorisation process

OAuth2 (pre-step)

The main purpose of this flow is to authenticate the PSU to get access to the services displayed by its ASPSP through the use of an access token obtained after the application of this protocol.

In order to simplify, the detail of this flow has been omitted from the Figure 4 Figure 3: Start of payment with OAuth2 as pre-step and SCA flow by redirection and can be found in the section 6.16.1 OAuth2 as a pre-step.VERIFY

Note: this step is optional. Only applies if no valid access token is available.

PSU initiates payment

The PSU wants to pay through the TPP.

1. Start Payment Request (TPP → ASPSP)

The TPP sends a POST request to initiate payment with *token_{HUB}* to HUB. Among the data reported by the TPP are:

- **TPP data:** identifier, name, roles, NCA, certificate ...
- **Payment data:** transfer type, ordering IBAN, beneficiary IBAN, amount, currency, concept ...
- **Data for risk scoring calculation:** IP, port, user-agent, language, location, HTTP headers ...
- **X-Request-ID:** identifier of the operation assigned by the TPP.
- **Access token** from TPP to Hub
- **TPP-Redirect-Preferred:** true (SCA flow preference by redirection) or not reported (ASPSP decides SCA by redirection).
- **TPP-Redirect-URI:** Return URI of the TPP after redirection to SCA.
- **TPP-Explicit-Authorization-Preferred:** false - TPP preference to initiate authorization implicitly
- **Other data**

4. Start Payment Response (ASPSP → TPP)

The Hub, after receiving the response from the ASPSP, responds to the TPP indicating that strong authentication (SCA) is required by redirecting to the authentication endpoint of the Hub, returning:

- **transactionStatus:** ISO 20022 state of the received payment start.
- **paymentId:** resource identifier generated by the Hub referring to the current payment initiation transaction.
- **_links**
 - **scaRedirect:** links to the Hub endpoint where after receiving the redirect from the TPP it redirects back to the scaRedirect of the ASPSP. This URL can add security parameters to allow session maintenance during redirection.

`https://hub.example.com/auth`
 - **self:** link to the payment resource generated by the Hub for the payment initiation request received from the TPP.
 - **state:** link of the Hub to which the TPP can make a request to check the state of the payment.
- **Other data**

3. TPP generates state

The TPP, after receiving the response to initiate payment, generates a value for *state* (XSRF token) that it must link to the PSU browser session.

4. Redirect to scaRedirect (TPP → ASPSP)

The TPP redirects the PSU to the authentication endpoint by adding to it the field *state* as a query-param.

```
HTTP/1.1 302 Found
Location: https://hub.example.com/auth?state=qwerty
```

SCA entre PSU ↔ ASPSP

During this redirection process, the ASPSP will be able to:

- Show consent to the PSU to access the available accounts
- Show available accounts and the PSU selects one of them
- Show commissions to the PSU if required
- Show ASPSP-PSU interface for SCA

6. SCA & Commissions

The ASPSP, after receiving the risk scoring of the operation, decides if SCA is necessary and executes it, showing the commissions.

Note: if the SCA process runs correctly, the payment is started.

11. Payment State Request (TPP → ASPSP)

The TPP will send a request for payment state with *token* to know the state of the payment.

12. Payment State Response (ASPSP → TPP)

The ASPSP updates the state of the operation and responds to the TPP.

7.2.1.1 SCA flow by redirection: explicit start of authorization process

Similar to 6.3.1.2 SCA flow by redirection: explicit start of authorization.

7.2.2 Payment initiation completion

This message is sent by the TPP to the Hub to initiate a payment without informing the issuer's account.

7.2.2.1 Request

Endpoint

POST {provider}/{aspsp}/v1.1/sva/payments/{payment-product}

Path

Field	Description	Type	Mandat.	Format
provider	URL of the HUB where the service is released.	String	MA	Ex: www.hub.com
aspsp	Name of the ASPSP to which the request is to be made.	String	MA	Ex: aspsp-name
payment-product	Paid product to use. List of supported products: <ul style="list-style-type: none"> • sepa-credit-transfers • instant-sepa-credit-transfers • target-2-payments • cross-border-credit-transfers 	String	MA	Ex: {provider}/{aspsp}/v1.1/payments/sepa-credit-transfers/

Header

The same as those defined in the section 6.3.2.1

Body

The content of the Body is defined in 8.14 SinglePayment.

7.2.2.2 Response

HTTP Code

201 if the resource has been created

Header

The same as those defined in the section 6.3.2.2

Body

The same as those defined in the section 6.3.2.2

7.2.2.3 Examples

Example of request

POST <https://www.hub.com/aspsp-name/v1.1/sva/payments/sepa-credit-transfers>

Content-Encoding: gzip

Content-Type: application/json

X-Request-ID: 10391c7e-ad88-49ec-a2ad-00aacb1f6541

Authorization: Bearer 2YotnFZFEjrlzCsicMWpAA

PSU-IP-Address: 192.168.8.16

PSU-IP-Port: 443

PSU-Accept: application/json

PSU-Accept-Charset: utf-8

PSU-Accept-Encoding: gzip

PSU-Accept-Language: es-ES

PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0) Gecko/20100101 Firefox/54.0

PSU-Http-Method: POST

PSU-Device-ID: f8b3feda-6fe3-11e8-adc0-fa7ae01bbebc

PSU-GEO-Location: GEO:12.526347;54.649862

TPP-Redirect-Preferred: true

TPP-Redirect-URI: https://www.tpp.com/cb

TPP-Nok-Redirect-URI: https://www.tpp.com/cb/nok

Date: Sun, 26 Sep 2017 15:02:37 GMT

```
{
  "instructedAmount": {
    "currency": "EUR",
    "amount": "153.50"
  },
  "creditorAccount": {
```

```

      "iban": "ES22222222222222222222222222222222",
    },
    "creditorName": "Nombrel23",
    "remittanceInformationUnstructured": "Información adicional"
  }
}

```

Example response

HTTP/1.1 201 Created

X-Request-ID: 10391c7e-ad88-49ec-a2ad-00aacb1f6541

ASPSP-SCA-Approach: REDIRECT

Date: Sun, 26 Sep 2017 15:02:43 GMT

Location: /v1.1/payments/sepa-credit-transfers/1234-qwer-5678

```
{
  "transactionStatus": "RCVD",
  "paymentId": "123-qwe-456",
  "_links": {
    "scaRedirect": {
      "href": "https://www.hub.com/aspsp-name/authorize"
    },
    "self": {
      "href": "/v1.1/payments/sepa-credit-transfers/123-qwe-456",
      "state": {
        "href": "/v1.1/payments/sepa-credit-transfers/123-qwe-456/state"
      }
    }
  }
}
```

7.3 SVA: Start of standing orders for recurring / periodic payments with list of accounts available for PISP

This service allows the TPP to initiate a payment without informing the issuer's account "debtorAccount" and provides the list of accounts during the SCA flow for the PSU to select one.

This valuable service complements the payments API and makes use of CORE services to:

- Obtain periodic payment state

- Retrieve recurring payment initiation information
- Cancel start of recurring payment

7.3.1 Periodic Payment Initiation Flows

7.3.1.1 SCA flow by redirection with account selection: implicit start of authorization process

The following represents the sequence of requests / responses that are necessary for this service.

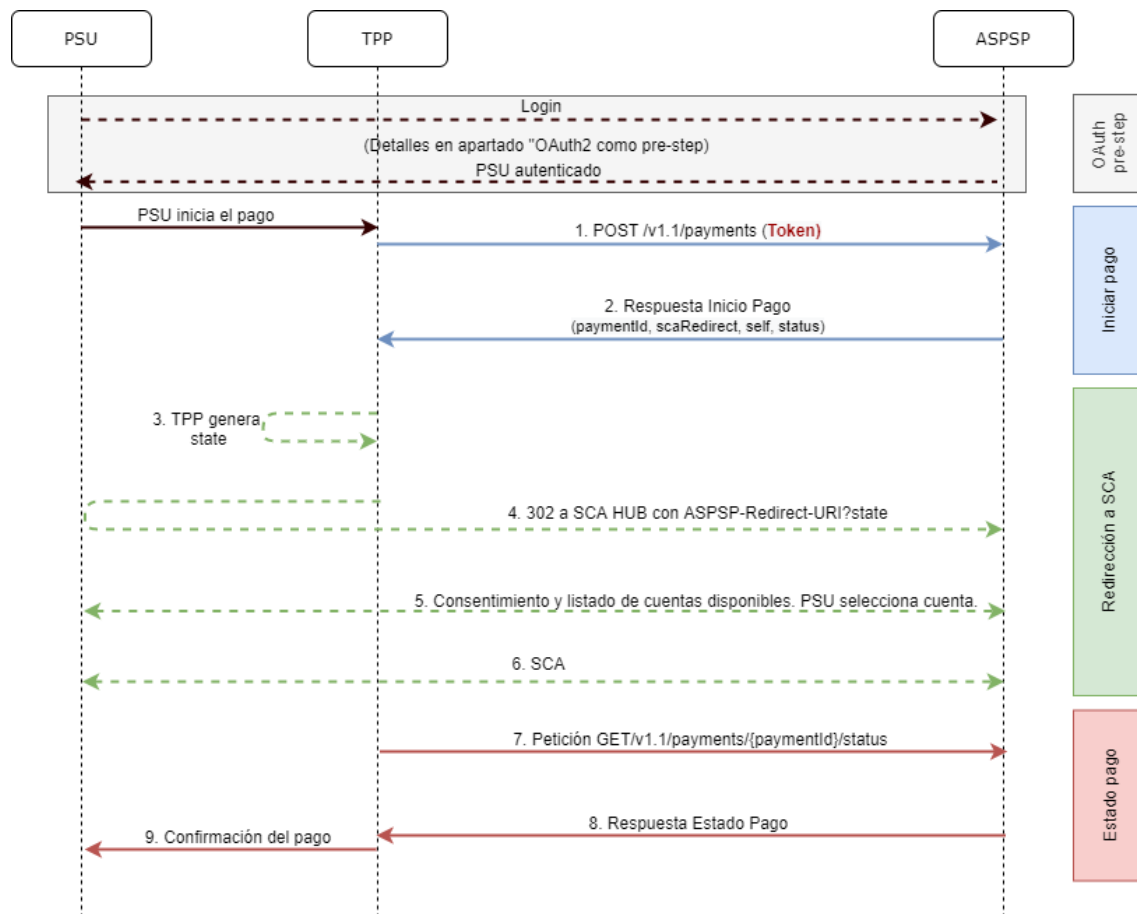


Figure 9: Scenario of starting periodic payment with list of accounts available for PISP

OAuth2 (pre-step)

The main purpose of this flow is to authenticate the PSU to get access to the services displayed by its ASPSP through the use of an access token obtained after the application of this protocol.

For simplicity, the detail of this flow has been omitted and can be found in section 6.16.1 OAuth2 as a pre-step.

Note: this step is optional. Only applies if no valid access token is available.

PSU starts periodic payment

The PSU wants to pay through the TPP.

1. Request Start Periodic Payment (TPP → Hub)

The TPP sends a POST request to initiate periodic payment with *tokenTPP* to the Hub. Among the data reported by the TPP are:

- **TPP data:** identifier, name, roles, NCA, certificate ...
- **Recurring payment details:** type of transfer, IBAN beneficiary, amount, currency, concept...
- **Data for risk scoring calculation:** IP, port, user-agent, language, location, HTTP headers ...
- **X-Request-ID:** identifier of the operation assigned by the TPP.
- **Access token** from the TPP

2. Request for Start Periodic Payment (Hub → ASPSP)

The Hub receives the request from the TPP, stores it and sends a POST request to initiate periodic payment with *tokenHUB* to the ASPSP. Among the data reported by the Hub are:

- **TPP data:** identifier, name, roles, NCA
- **Periodic payment data:** transfer type, beneficiary IBAN, amount, currency, concept, periodic payment start date, periodicity, ...
- **X-Request-ID:** identifier of the operation assigned by the TPP.
- **HUB-Transaction-ID:** Hub operation identifier
- **HUB-Request-ID:** Hub request identifier
- **Access token** from the TPP

3. Response Start Periodic Payment (ASPSP → Hub)

The ASPSP returns to the Hub a link to *scaRedirect* where the accounts available to the PSU will be displayed:

-
- **transactionStatus**: ISO 20022 state of the start of the periodic payment received.
 - **paymentId**: resource identifier generated by the ASPSP that refers to the current recurring payment initiation transaction.
 - **_links**
 - **scaRedirect**: link to the ASPSP authentication server where the accounts available to the PSU will be displayed and to initiate SCA via a redirect (SCA over OAuth2 does not apply). This URL can add security parameters to allow session maintenance during redirection.

`https://aspsp.example.com/auth`
 - **self**: link to the payment resource generated by the ASPSP for the payment initiation request received from the TPP.
 - **state**: ASPSP link to which the Hub may make a payment state query request.
 - Other information regarding the operation.

4. Response Start Periodic Payment (Hub → TPP)

The Hub, after receiving the response from the ASPSP, responds to the indicated TPP the url to which it has to redirect to continue with the process:

- **transactionStatus**: ISO 20022 state of the start of the periodic payment received.
- **paymentId**: resource identifier generated by the Hub referring to the current periodic payment initiation transaction.
- **_links**
 - **scaRedirect**: links to the Hub endpoint where after receiving the redirect from the TPP it redirects back to the scaRedirect of the ASPSP. This URL can add security parameters to allow session maintenance during redirection.

`https://hub.example.com/auth`
 - **self**: link to the payment resource generated by the Hub for the payment initiation request received from the TPP.
 - **state**: link of the Hub to which the TPP can make a request to check the state of the payment.
- Other data regarding the operation.

5. Redirection to scaRedirect from the Hub (TPP → Hub)

The TPP, after receiving the response to initiate periodic payment, redirects the PSU to the authentication endpoint of the Hub.

HTTP/1.1 302 Found
Location: <https://hub.example.com/auth>

6. Redirect to scaRedirect from ASPSP (→ASPSP Hub)

The Hub, upon receiving the redirection from the TPP, will redirect to the ASPSP authentication server where the accounts available to the PSU will be displayed and, if the ASPSP considers it, it will trigger strong authentication (SCA).

HTTP/1.1 302 Found
Location: <https://aspsp.example.com/auth>

SCA entre PSU ↔ ASPSP

During this redirection process, the ASPSP will be able to:

- Show consent to the PSU to access the available accounts
- Show available accounts and the PSU selects one of them
- Show commissions to the PSU if required
- Show ASPSP-PSU interface for SCA

SCA & Commissions

The ASPSP, after receiving the risk scoring of the operation, decides if SCA is necessary and executes it, showing the commissions.

Note: if the SCA process runs correctly, the payment is started.

9. Redirection to Hub URL (ASPSP → Hub)

After redirection to the SCA in the ASPSP environment, it will return control to the Hub.

HTTP/1.1 302 Found
Location: <https://hub.example.com/cb?state=xyz>

10. Redirection to TPP URL (→TPP Hub)

The Hub, after receiving the redirection back from the ASPSP at the end of the SCA, redirects to the *callback* URL of the TPP to return control.

HTTP/1.1 302 Found
Location: <https://tpp.example.com/cb>

11. Request State Periodic Payment (TPP → Hub)

The TPP will send a payment state request with *tokenTPP* to the Hub for payment state.

12. Periodical Payment State Request (Hub → ASPSP)

The Hub will relay the payment state request with *tokenHUB* to the ASPSP to know the state of the payment.

Note: the Hub performs an exchange between *tokenTPP* and *tokenHUB*.

13. Response State Periodic Payment (ASPSP → Hub)

After receiving the periodic payment state request with valid *tokenHUB*, the ASPSP checks the state of the payment initiation in its systems and returns it to the Hub.

14. State Response Periodic Payment (→TPP Hub)

The Hub after receiving the response from the ASPSP, updates the state of the operation and responds to the TPP.

Confirmation of periodic payment

The TPP confirms the state of the payment to the PSU.

7.3.1.2 SCA flow by redirection: explicit start of authorization process

Similar to 6.3.1.2 SCA flow by redirection: explicit start of authorization.

7.3.2 Payment initiation completion

Message sent by the TPP to the ASPSP through the Hub to create a recurring / periodic payment start without informing the issuer's account "debtorAccount".

A TPP can send a recurring payment start where the start date, frequency and, conditionally, end date are provided.

Once authorized by the PSU, the payment will be executed by the ASPSP, if possible, following the "standing order" as it was sent by the TPP. No further action is required from the TPP.

In this context, this payment is considered a periodic payment to differentiate the payment from other types of recurring payments where third parties are initiating the same amount of money.

Reglas campo dayOfExecution

- **Monthly payments or higher:** possible values range from 01 to 31. Using 31 as the last day of the month

7.3.2.1 Request

Endpoint

POST {provider}/v1.1/sva/periodic-payments/{payment-product}

Path

Field	Description	Type	Mandat.	Format
provider	URL of the ASPSP where the service is published.	String	MA	Ex: aspsp.example.es
payment-product	Paid product to use. List of supported products: <ul style="list-style-type: none"> • sepa-credit-transfers • instant-sepa-credit-transfers • target-2-payments • cross-border-credit-transfers 	String	MA	Ex: {provider}/v1.1/periodic-payments/sepa-credit-transfers/

Query parameters:

No additional parameters are specified for this request.

Header

The same as those defined in the section 6.3.2.1

Body

The content of the Body is defined in section 6.3.2.1, following the conditions of these tables, plus those defined below:

Field	Description	Type	Mandat	Format
startDate	The first applicable day of execution from this date is the first payment	String	MA	ISODate xEx: "startDate":"2018-12-20"
executionRule	Supported values: <ul style="list-style-type: none">• following• preceding Defines the behavior when recurring payment dates fall on weekends or holidays. Payment is then executed on the preceding or following working day.	String	OP	Ex: "executionRule":"following"

	The ASPSP may reject the request due to the communicated value if the Online Banking rules do not support this execution rule.			
endDate	The last applicable day of execution. If not given, it is an endless standing order.	String	OP	ISODate Ex: "endDate": "2019-01-20"
frequency	The frequency of the recurring payment resulting from this standing order. Allowed values: <ul style="list-style-type: none"> Monthly Quarterly Semi Annual 	String	MA	EventFrequency7Code de ISO 20022 Ex: "frequency": "Monthly"
dayOfExecution	"31" is last. Follows the regular expression <code>\d{1,2}</code> The date refers to the ASPSP time zone. Only if supported in ASPSP Online Banking.	String	COND	<code>\d{1,2}</code> Ex: "dayOfExecution": "01"

7.3.2.2 Response

HTTP Code

201 if the resource has been created

Header

The same as those defined in the section 6.3.2.2

Body

The same as those defined in the section 6.3.2.2

Body

Field	Description	Type	Mandat.	Format
transactionStatus	Transaction state.	String	MA	ISO 20022 Ex: "transactionStatus": "RCVD"
paymentId	Resource identifier that refers to the initiation of payment.	String	MA	^.{1,36}\$ Ex: "paymentId": "1b3ab8e8-0fd5-43d2-946e-d75958b172e7"
transactionFees	Commissions associated with payment.	Amount	OP	Ex: "transactionFees": {...}
transactionFeeIndicator	If equal to "true", the transaction will incur a commission according to the ASPSP or as agreed between ASPSP and PSU.	Boolean	OP	Ex: "transactionFeeIndicator": true

PSD2 - APIs Implementation Guide v1.1 for TPPs

	If it is equal to "false" or not used, the transaction will not involve any additional fees for the PSU.			
_links	<p>List of hyperlinks to be recognized by the TPP. Supported types in this response:</p> <ul style="list-style-type: none"> • scaRedirect: in case of SCA by redirection. Link where the PSU browser must be redirected by the Hub. • scaOAuth: in case of SCA and require payment execution. • self: link to the 	Links	MA	Ex: "_links": {...}

	<p>payment initiation resource created by this request.</p> <ul style="list-style-type: none"> state: link to retrieve the state of the payment initiation transaction. 			
psuMessage	Text sent to the TPP through the HUB to be displayed to the PSU.	String	OP	$\wedge.\{1,500\}$ \$ Ex: "psuMessage": "Información para PSU"
tppMessages	Message for the TPP sent through the HUB.	List<TppMessage>	OP	Ex: "tppMessages": [...]

7.3.2.3 Examples

Example of request

POST <https://aspsp.example.es/v1.1/sva/periodic-payments/sepa-credit-transfers>

Content-Encoding: gzip

Content-Type: application/json

PSD2 - APIs Implementation Guide v1.1 for TPPs

HUB-Transaction-ID: 3dc3d5b3-7023-4848-9853-f5400a64e80f
HUB-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7721
X-Request-ID: 10391c7e-ad88-49ec-a2ad-00aacb1f6541
TPP-HUB-ID: PSDES-BDE-3DFD21
TPP-HUB-Name: Nombre del TPP
TPP-HUB-Rol: PSP_PI
TPP-HUB-National-Competent-Authority: BDE
Authorization: Bearer 2YotnFZFEjrlzCsicMWpAA
PSU-IP-Address: 192.168.8.16
PSU-IP-Port: 443
PSU-Accept: application/json
PSU-Accept-Charset: utf-8
PSU-Accept-Encoding: gzip
PSU-Accept-Language: es-ES
PSU-User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0
PSU-Http-Method: POST
PSU-Device-ID: f8b3feda-6fe3-11e8-adc0-fa7ae01bbebc
PSU-GEO-Location: GEO:12.526347;54.649862
TPP-Redirect-Preferred: true
TPP-Redirect-URI: https://hub.example.es/cb
TPP-Nok-Redirect-URI: https://hub.example.es/cb/nok
Date: Sun, 26 Sep 2017 15:02:37 GMT
{
 "instructedAmount": {
 "currency": "EUR",
 "amount": "153.50"
 },
 "creditorAccount": {
 "iban": "ES22222222222222222222222222222222"
 },
 "creditorName": "Nombre123",
 "remittanceInformationUnstructured": "Información adicional",
 "startDate": "2018-03-01",
 "executionRule": "preceding",

PSD2 - APIs Implementation Guide v1.1 for TPPs

```
"frequency": "Monthly",  
"dayOfExecution": "01"  
}
```

Example response

```
HTTP/1.1 201 Created  
HUB-Transaction-ID: 3dc3d5b3-7023-4848-9853-f5400a64e80f  
HUB-Request-ID: 99391c7e-ad88-49ec-a2ad-99ddcb1f7721  
X-Request-ID: 10391c7e-ad88-49ec-a2ad-00aacb1f6541  
ASPS-SCA-Approach: REDIRECT  
Date: Sun, 26 Sep 2017 15:02:43 GMT  
Location: https://aspsp.example.es/v1.1/periodic-payments/123-qwe-456  
Content-Type: application/json  
{  
  "transactionStatus": "RCVD",  
  "paymentId": "123-qwe-456",  
  "_links": {  
    "scaRedirect": {  
      "href": "https://aspsp.example.es/authorize"  
    },  
    "self": {  
      "href": "/v1.1/periodic-payments/123-qwe-456",  
      "state": {  
        "href": "/v1.1/periodic-payments/123-qwe-456/state"  
      }  
    }  
  }  
}
```

8. DEFINITION OF TYPES OF COMPOSITE DATA

The following defines the composite data types used in system requests and responses.

8.1 AccountAccess

Field	Description	Type	Mand at.	Format
accounts	Indicates the accounts on which to request detailed information. If the list is empty, the TPP is requesting all accessible accounts and will be asked in a PSU-ASPSP dialogue. In addition, the list of balances and transactions must also be empty if they are used.	List<AccountReference>	OP	Ex: "accounts": [...]
balances	Indicates the accounts on which to request balances. If the list is empty, the TPP is requesting all accessible accounts and will be asked in a PSU-ASPSP dialogue. In addition, the list of balances and transactions must also be empty if they are used.	List<AccountReference>	OP	Ex: "balances": {...}
transactions	Indicates the accounts on which to request transactions.	List<AccountReference>	OP	Ex: "transactions": {...}

	If the list is empty, the TPP is requesting all accessible accounts and will be asked in a PSU-ASPSP dialogue. In addition, the list of balances and accounts must also be empty if they are used.			
additionalInformation	Note: the information contained in this object will be ignored by the ASPSP.	Additional InformationAccess	OP	Ex: "additionalInformation": {...}
availableAccounts	Only the value "allAccounts" is allowed	String	OP	Ex: "availableAccounts": "allAccounts"
availableAccountsWithBalance	Only the value "allAccounts" is allowed	String	OP	Ex: "availableAccountsWithBalance": "allAccounts"
allPsd2	Only the value "allAccounts" is allowed	String	OP	Ex: "allPsd2": "allAccounts"

8.2 AccountDetails

Field	Description	Type	Mandat.	Format
resourceId	Identifier of the account to be used in the PATH when requesting data about a dedicated account.	String	COND	^. {1,100} \$ Ex: "resourceId": "3dc3d5b3702348489853f5400a64e80f"
iban	Account IBAN	String	OP	Ex: "iban": "ES11111111111111111111"

PSD2 - APIs Implementation Guide v1.1 for TPPs

bban	BBAN of the account, when it does not have an IBAN.	String	OP	Ex: "bban": "20385778983000760236"
msisdn	Alias to access a payment account via a registered mobile phone number.	String	OP	^. {1,35}\$ Ex: "msisdn": "..."
currency	Account Currency	String	MA	ISO 4217 Ex: "currency": "EUR"
ownerName	Name of the legal owner of the account (in this case, the name of the connected PSU. For a corporate account, the corporate name will be used in this field.	String	OP	^. {1,140}\$ Ex: "ownerName": "Heike Mustermann"
name	Account name assigned by ASPSP in agreement with the account owner in order to provide a new way to identify the account.	String	OP	^. {1,70}\$ Ex: "name": "Name assigned by the ASPSP"
displayName:	Name of the account defined by the PSU in the Online Channels	String	OP	^. {1,70}\$ Ex: "displayName": "Name assigned by the PSU"
product	Product name that the ASPSP gives to this account.	String	OP	^. {1,35}\$ Ex: "product": "Main Account"
cashAccountType	Specify the nature or use of the account.	String	OP	ExternalCashAccountType1Code de ISO 20022 Ex: "cashAccountType": "CACC"

PSD2 - APIs Implementation Guide v1.1 for TPPs

state	State of the account. The value is one of the following: <ul style="list-style-type: none"> enabled: account is available deleted: account closed blocked: account blocked 	String	OP	Ex: "state": "enabled"
bic	BIC associated with the account.	String	OP	^.{1,12} \$ Ex: "bic": "XSHXSMXXX"
linkedAccounts	In this field the ASPSP can name an account associated with pending card transactions.	String	OP	^.{1,70}\$
usage	Specifies the use of the account. Possible values: <ul style="list-style-type: none"> PRIV: private personal account ORGA: professional account 	String	OP	^.{1,4}\$ Ex: "usage": "PRIV"
details	Specifications that must be provided by the ASPSP. <ul style="list-style-type: none"> Features of the account Characteristics of the card 	String	OP	^.{1,500} \$
balances	Account balances.	List<Balances>	COND	"balances": [...]

_links	<p>Links to the account to retrieve account balance and/or transaction information.</p> <p>Links supported only when the corresponding consent to the account has been given.</p>	Links	OP	Ex: "links": {...}
---------------	---	-------	----	--------------------

8.3 AccountReference

Field	Description	Type	Mand at.	Format
iban	Account IBAN	String	COND	Ex: "iban": "ES11111111111111111111"
bban	BBAN of the account, when it does not have an IBAN.	String	COND	Ex: "bban": "20385778983000760236"
pan	Primary Account Number of the card. It can be tokenized by the ASPSP to meet PCI DSS requirements.	String	COND	^. {1,35}\$ Ex: "pan": "1234567891234567"
maskedPan	Primary Account Number of the card in masked form.	String	COND	^. {1,35}\$ Ex: "maskedPan": "123456*****4567"
msisdn	Alias to access a payment account via a registered mobile phone number.	String	COND	^. {1,35}\$ Ex: "msisdn": "..."
currency	Currency	String	OP	ISO 4217 Ex: "currency": "EUR"

8.4 AccountReport

Field	Description	Type	Mand at.	Format
booked	Last known transactions (annotations) of the account. It must be included if the bookingStatus parameter is set to "booked" or "both".	List<Trans actions>	COND	Ex: "booked":[{..}]
pending	Pending account transactions. Not contained if the bookinStatus parameter is set to "booked".	List<Trans actions>	OP	Ex: "pending":[{..}]
informatio n	List of standing orders Included if the bookingStatus parameter is set to "information".	List<Trans actions>	OP	Ex: "information": [{...}]
_links	The following links are accepted in this object: <ul style="list-style-type: none"> • account (OB) • first (OP) • next (OP) • previous (OP) • last (OP) 	Links	MA	Ex: "_links":[{..}]

8.5 AdditionalInformationAccess

Field	Description	Type	Mand at.	Format
-------	-------------	------	----------	--------

trustedBeneficiaries	It is requesting access to the trusted payees of the referenced and PSU-related account. Note: if reported it will be ignored by the ASPSP.	List<AccountReference>	OP	Ex: "trustedBeneficiaries": {...}
ownerName	Note: if reported it will be ignored by the ASPSP	List<AccountReference>	OP	Ex: "ownerName": {...}

8.6 Address

Field	Description	Type	Mandat.	Format
streetName	Street	String	OP	^.{1,70}\$ Ex: "street": "Street example"
buildingNumber	Number	String	OP	Ex: "buildingNumber": "5"
townName	City	String	OP	Ex: "city": "Cordoba"
PostCode	Postcode	String	OP	Ex: "postalCode": "14100"
country	Country code	String	MA	ISO 3166 Ex: "country": "ES"

8.7 Amount

Field	Description	Type	Mandat.	Format
currency	Currency of the amount.	String	MA	ISO 4217 Ex: "currency": "EUR"
amount	Amount The decimal separator is the period.	String	MA	ISO 4217 Ex:

				"amount": "500.00"
--	--	--	--	--------------------

8.8 AuthenticationObject

Field	Description	Type	Mand at.	Format
authenticationType	Authentication method type. Possible values: <ul style="list-style-type: none"> • SMS_OTP • CHIP_OTP • PHOTO_OTP • PUSH_OTP See annex 9.6 iError! No se encuentra el origen de la referencia. iError! No se encuentra el origen de la referencia. for more information.	String	MA	Ex: "authenticationType": "SMS_OTP"
authenticationVersion	Version of the tool associated with the authenticationType.	String	COND	Ex: "authenticationVersion": "1.0"
authenticationMethodId	ID of the authentication method provided by the ASPSP.	String	MA	^.{1,35}\$
name	Name of the authentication method defined by the PSU in ASPSP online banking. Alternatively, it could be a description provided by the ASPSP.	String	MA	Ex: "name": "SMS OTP al teléfono 666777888"

	If the TPP has it available, it must be presented to the PSU.			
explanation	Detailed information about the SCA method for the PSU	String	OP	

8.9 Aspsp

Field	Description	Type	Mandat	Format
bic	ASPSP BIC code.	String	MA	Ex: "bic": "XXXXXXXXXX"
name	ASPSP Name	String	OP	Ex: "name": "Nombre ASPSP"
apiName	Name of the ASPSP used in the request PATH. Note: Only available for V2 from the list of available ASPSPs.	String	COND	Ex: "apiName": "nombreBanco"

8.10 Balance

Field	Description	Type	Mandat.	Format
balanceAmount	Balance amount and currency	Amount	MA	Ex: "balanceAmount": {...}
balanceType	Balance type. Values supported in annex 9.6 Types of balances	String	MA	Ex: "balanceType": "closingBooked"

creditLimitIncluded	Flag indicating if the credit limit of the corresponding account is included in the balance calculation, when applicable.	Boolean	OP	Ex: "creditLimitIncluded":true
lastChangeDateTime	Date of the last action carried out on the account.	String	OP	ISODateTime Ex: "lastChangeDateTime": "2017-10-25T15:30:35.035Z"
referenceDate	Balance sheet reference date	String	OP	ISODate Ex: "referenceDate": "2017-10-25"
lastCommittedTransaction	entryReference of the last transaction to help the TPP identify whether all PSU transactions are already known.	String	OP	Max35Text Ex: "lastCommittedTransaction": "1234-asd-567"

8.11 ExchangeRate

Field	Description	Type	Mandat.	Format
currencyFrom	Original currency	String	MA	Ex: "currencyFrom": "USD"
rate	Define the interchange fee. Ex: currencyFrom=USD, currencyTo=EUR: 1USD =0.8 EUR and 0.8 is the fee.	String	MA	Ex: "rate": "0.8"
currencyTo	Currency of destination	String	MA	Ex: "currencyTo": "EUR"
rateDate	Rate date	String	MA	ISODateTame
rateContract	Reference to the rate contract	String	OP	

8.12 Href

Field	Description	Type	Mand at.	Format
href	Contains a link to a resource	String	OP	Ex: "href": "/v1.1/payments/sepa-credit-transfers/asd-1234-jkl"

8.13 Links

Field	Description	Type	Mand at.	Format
scaRedirect	URL used to perform SCA, by redirection of the PSU browser.	Href	OP	Ex: "scaRedirect": {...}
scaOAuth	The link to retrieve a JSON document that specifies the ASPSP authorization server details. The JSON document follows the definition given at https://tools.ietf.org/html/draft-ietf-oauth-discovery . Only for ASPSPs that require Payment Execution.	Href	OP	Ex: "scaOAuth": {...}
startAuthorisation	Link to the endpoint where the authorization of the transaction or the authorization of the cancellation transaction must be initiated.	Href	OP	Ex: "startAuthorisation": {...}

startAuthorisationWithAuthenticationMethodSelection	Link to the endpoint where the authorization of a transaction or a cancellation transaction must be initiated, where the SCA method must be informed with the corresponding call.	Href	OP	Ex: "startAuthorisationWithAuthenticationMethodSelection": {...}
selectAuthenticationMethod	Link where the TPP can select the applicable 2nd factor authentication method for the PSU, in case there is more than one.	Href	OP	Ex: "selectAuthenticationMethod": {...}
self	The link to the resource created for the request. This link can later be used to retrieve the state of the transaction.	Href	OP	Ex: "self": {...}
state	The link to retrieve the state of the transaction. For example, payment start state.	Href	OP	Ex: "state": {...}
scaStatus	Link to retrieve the state of the authorization or cancellation authorization sub-resource.	Href	OP	Ex: "scaStatus": {...}
account	Link to the resource that provides the data of an account.	Href	OP	Ex: "account": {...}
balances	Link to the resource that provides the account balances.	Href	OP	Ex: "balances": {...}
transactions	Link to the resource that provides the account transactions.	Href	OP	Ex: "transactions": {...}

transactionDetails	Link to resource providing details of a specific transaction. NOT SUPPORTED IN THIS VERSION.	Href	OP	
first	Navigation link for paginated account reports.	Href	OP	Ex: 'first': {...}
next	Navigation link for paginated account reports.	Href	OP	Ex: 'next': {...}
previous	Navigation link for paginated account reports.	Href	OP	Ex: 'previous': {...}
last	Navigation link for paginated account reports.	Href	OP	Ex: "last": {...}
download	Download link for large AIS data packets. Only for camt-data.	Href	OP	Ex: "download": {...}

8.14 SinglePayment

Field	Description	Type	Format
endToEndIdentification	Unique identifier of the operation assigned by the initiating party (TPP)	String	^. {1,35}\$ Ex: "endToEndIdentification": "12345678901234567890123456789012345"
debtorName	Issuer's name	String	^. {1,70}\$ Ex: 'debtorName': 'John Doe'
debtorAccount	Issuer's account. Note: this field may be optional in some services such as bulk payments	Account Reference	Ex: "debtorAccount": {"iban": "ES11111111111111111111111111111111"}
instructedAmount	Information on the transfer made.	Amount	Ex: "instructedAmount": {...}

creditorAccount	Beneficiary Account	Account Reference	Ex: `creditorAccount`: {`iban`: `ES111111111111111111`}
creditorAgent	BIC of the beneficiary's account.	String	Ex: `creditorAgent`: `XSHXSMXX`
CreditorName	Beneficiary name	String	^. {1,70}\$ Ex: `creditorName`: `Name`
creditorAddress	Beneficiary Address	Address	Ex: `creditorAddress`: {...}
chargeBearer	Only for payment-product: <ul style="list-style-type: none"> target-2-payments cross-border-credit-transfers Allowed values: <ul style="list-style-type: none"> DEBT CRED SHAR SLEV 	String	ChargeBearerType1Code from ISO 20022 Ex: `chargeBearer`: `SLEV`
remittanceInformationUnstructured	Additional information: See Annex Error! No se encuentra el origen de la referencia. No se encuentra el origen de la referencia. for recommendations for use.	String	^. {1,140}\$ Ex: `remittanceInformationUnstructured`: `Additional information`
requestedExecutionDate	Execution date	String	ISODate Ex: "requestedExecutionDate": "2018-05-17"
requestedExecutionTime	Date/time executed	String	ISODateTime

8.15 StandingOrderDetails

Field	Description	Type	Mandat.	Format
startDate	The first applicable day of execution from this date is the first payment	String	MA	ISODate xEx: "startDate":"2018-12-20"
endDate	The last applicable day of execution. If not given, it is an endless standing order.	String	OP	ISODate Ex: "endDate":"2019-01-20"
executionRule	Supported values: <ul style="list-style-type: none"> • following • preceding <p>Defines the behavior when recurring payment dates fall on weekends or holidays. Payment is then executed on the preceding or following working day.</p> <p>The ASPSP may reject the request due to the communicated value if the Online Banking rules do not support this execution rule.</p>	String	OP	Ex: "executionRule":"following"
withinAMonthFlag	This element is only used in case the frequency is equal to "monthly". If this element is equal to false, it has no effect.	Boolean	OP	Ex: 'withinAMonthFlag': true

	<p>If it is equal to true, then the execution rule is overridden if the execution day falls in a different month.</p> <p>Note: this attribute is rarely used.</p>			
frequency	<p>The frequency of the recurring payment resulting from this standing order.</p> <p>Allowed values:</p> <ul style="list-style-type: none"> • Monthly • Quarterly • Semi Annual 	String	MA	<p>EventFrequency7Code de de ISO 20022</p> <p>Ex: "frequency": "Monthly"</p>
monthsOfExecution	<p>Following the regular expression \d{1,2} The array is restricted to 11 entries. The values contained in the array must all be different, and the maximum value of an input is 12.</p> <p>The attribute is contained if and only if the frequency equals "MonthlyVariable".</p>	List<String>	COND	<p>Ex: "monthsOfExecution": ["1", "4", "10"]</p>
multiplier	<p>Frequency multiplier. For example, frequency = weekly and multiplier = 3 means every 3 weeks.</p> <p>Note: this attribute is rarely used</p>	Integer	OP	<p>Ex: "multiplier": 3</p>
dayOfExecution	<p>"31" is last.</p> <p>Following the regular expression \d{1,2}</p>	String	COND	<p>\d{1,2}</p> <p>Ex: "dayOfExecution": "01"</p>

	The date refers to the ASPSP time zone. Only if supported in ASPSP Online Banking.			
limitAmount	Limit amount for Restrictions: transactionAmount must be zero and bankTransactionCode must specify PMNT-MCOP-OTHR for	Amount	COND	Ex: "limitAmount": {...}

8.16 StructuredAdditionalInformation

Field	Description	Type	Mandat.	Format
standingOrderDetails	Standing order details	Standing OrderDetails	MA	Ex: "standingOrderDetails": {...}

8.17 TppMessage

Field	Description	Type	Mandat.	Format
category	Category of the type of message received. Possible values: ERROR or WARNING	String	MA	Ex: "category": "ERROR"
code	Response code.	String	MA	Ex: "code": "CONSENT_INVALID"

	All return codes by service 9.3 Return Codes are listed in the annex 9.3.			
path	Path to the field referencing the error.	String	COND	Ex: "path": "..."
text	Additional explanatory text.	String	OP	Ex: "text": "Text example"

8.18 Transactions

Field	Description	Type	Mandat.	Format
transactionId	It can be used as access-ID in the API, where more details about the transaction can be offered. If this data is provided, the request for transaction details can be accessed.	String	OP	Ex: "transactionId": "123-asdf-456"
entryReference	Identification of the transaction that can be used, for example, in delta queries.	String	OP	^. {1,35}\$ Ex: "entryReference": "1234-asdf-456"
endToEndId	Unique end to end identifier.	String	OP	^. {1,35}\$ Ex: "endToEnd": "..."
mandateId	Identification of the mandate. For example, an ID from a SEPA mandate.	String	OP	^. {1,35}\$ Ex: "mandateId": "..."
checkId	Check identifier	String	OP	^. {1,35}\$ Ex: "checkId": "..."
creditorId	Beneficiary ID For example, a SEPA Payee ID.	String	OP	^. {1,35}\$ Ex: "creditorId": "..."

PSD2 - APIs Implementation Guide v1.1 for TPPs

bookingDate	Date of entry of the transaction	String	OP	ISODate 'bookingDate': ' 2017-10-23'
"valueDate": ""	Date on which the settlement becomes available to the account owner in the event of a credit.	String	OP	ISODate Ex: 'valueDate': ' 2017-10-23 '
transactionAmount	Transaction amount	Amount	MA	Ex: 'transactionAmount': [{... }]
currencyExchange	EXCHANGE RATE	List<ReportExchangeRate>	OP	Ex: 'currencyExchange': [{... }]
CreditorName	Name of the payee if the transaction is a charge.	String	OP	^. {1,70}\$ Ex: 'creditor': 'Name'
creditorAccount	Beneficiary Account	AccountReference	COND	Ex: 'creditorAccount': { ... }
creditorAgent	BIC of the beneficiary's account	String	OP	Ex: 'creditorAgent': 'XXXSDH'
ultimateCreditor	Last part they owe money to	String	OP	^. {1,70}\$ Ex: 'ultimateCreditor': 'Name'
debtorName	Name of the payer if the transaction is a credit.	String	OP	^. {1,70}\$ Ex: 'debtor': 'Name'
debtorAccount	Issuer's account.	AccountReference	COND	Ex: "debtorAccount": { ... }
debtorAgent	BIC associated with the issuing ASPSP	String	OP	Ex: "debtorAgent": "BIC"
ultimateDebtor	Last part owing an amount of money	String	OP	^. {1,70}\$ Ex: "ultimateDebtor": "Nombre"
remittanceInformationUnstructured	Field to include additional information about the sending.	String	OP	^. {1,140}\$ Ex: "remittanceInformationUnstructured": "Informacion adicional"

PSD2 - APIs Implementation Guide v1.1 for TPPs

remittanceInformationUnstructuredArray	Note: in version 2 of the standard the two remittanceUnstructured could be merged into one	List<String>	OP	\wedge . {1,140} \$ per String Ex: "remittanceInformationUnstructuredArray":["info1", "info2"]
remittanceInformationStructured	Field to include a reference to the sending.	String	OP	\wedge . {1,140} \$ Ex: "remittanceInformationStructured": "Ref. 12344567 "
remittanceInformationStructuredArray	Note: in version 2 of the standard the two remittanceUnstructured could be merged into one	List<String>	OP	\wedge . {1,140} \$ per String Ex: "remittanceInformationStructuredArray":["info1", "info2"]
additionalInformation	Used by the TPP to carry additional information from the PSU	String	OP	\wedge . {1,500} \$ Ex: "additionalInformation": "Información"
additionalInformationStructured	It is used only if the bookingStatus field contains the value "información". Each active standing order related to the payment account results in one entry.	StructuredAdditionalInformation	OP	Ex: "additionalInformationStructured": {...}
purposeCode	ExternalPurpose1Code ISO 20022	String	OP	ExternalPurpose1Code ISO 20022
bankTransactionCode	Bank transaction codes used by the ASPSP and using the sub-elements of the structured codes defined in ISO 20022. For standing orders the following codes apply:	String	OP	ExternalBankTransactionDomain1Code <ul style="list-style-type: none"> Ex: "bankTransactionCode": "PMNT-ICDT-STD0"

	<ul style="list-style-type: none"> • "PMNT-ICDT-STD0" for credit transfers, • "PMNT-IRCT-STD0" for instant credit transfers • "PMNT-ICDT-XBST" for cross-border credit transfers • "PMNT-IRCT-XBST" for cross-border real time credit transfers <p>"PMNT-MCOP-OTHR" for specific standing orders which have a dynamic amount when withdrawing funds. For example, at the end of the month to a savings account.</p> <p>This field is formed by concatenating the three ISO20022 codes:</p> <ul style="list-style-type: none"> • Domain • Family • Sub-family <p>Separated by "-"</p> <p>Example:</p> <p>PMNT-RCTD-ESCT defines a transaction assigned to the PayMeNT domain (PMNT), ReceivedCreDitTrans fer family (RCTD) and EuropeanSEPACredit Transfer (ESCT).</p>			
--	--	--	--	--

PSD2 - APIs Implementation Guide v1.1 for TPPs

proprietaryBankTransactionCode	Bank owner transaction code	String	OP	^.{1,35}\$
balanceAfterTransaction	Saldo después de la transacción. Recommended balance is interimBooked	Balance	OP	Ex: "balanceAfterTransaction": {...}
_links	Possible values: <ul style="list-style-type: none"> transactionDetails 	Links	OP	Ex: "_links": {...}

8.19 TrustedBeneficiary

Field	Description	Type	Mand at.	Format
trustedBeneficiaryId	Resource identifier of the entry in the list.	String	MA	UUID Ex: "trustedBeneficiaryId": "1b3ab8e8-0fd5-43d2-946e-d75958b172e7"
debtorAccount	Provided by the ASPSP if the trusted beneficiary entry is applicable only to a specific account.	AccountReference	OP	Ex: "debtorAccount": {...}
creditorAccount	Beneficiary Account	AccountReference	MA	Ex: "creditorAccount": {...}
creditorAgent	Mandatory when the information is mandatory for the related credit transfer. Eg. payments outside the SEPA zone.	String	COND	Ex: "creditorAgent": ""
CreditorName	Beneficiary name as provided by the PSU.	String	MA	Ex: "creditorName": "Beneficiary name"
creditorAlias	Alias defined by the PSU that is displayed in the list of trusted payees of the ASPSP online channels.	String	OP	Ex: "creditorAlias": "Alias"

9. ANNEXES

9.1 Signature

9.1.1 Header "Digest" required

The Digest field is mandatory in all requests.

This field contains a Hash of the body of the message. If the message does not contain a body, the "Digest" field must contain a hash of an empty "bytelist". The hashing algorithms that can be used to calculate the "Digest" in the context of this specification are SHA-256 and SHA-512.

9.1.2 Signature requirements

The structure of the "Signature" field in the request header must have the following structure:

Element	Type	Mandat.	Requirements	Additional requirements
KeyId	String	MA	It is a string that the HUB can use to find the component it needs to validate the signature.	Serial number of the TPP certificate included in "TPP-Signature-Certificate". It should be formatted as follows: KeyId="SN=XXX,CA=YYYYYYYYYYYYYYYY" Where "XXX" is the serial number of the certificate in hexadecimal encoding and "YYYYYYYYYYYYYYYY" is the complete "Distinguished Name" of the certificate authority.
Algorithm-ID	String	MA	It is used to specify the algorithm used to generate the signature.	The algorithm must identify the same algorithm for the signature as the one presented in the request certificate. It must identify SHA-256 or SHA-512.

PSD2 - APIs Implementation Guide v1.1 for TPPs

Header	String	OP	<p>It is used to specify the list of HTTP headers included when the signature for the message is generated.</p> <p>If specified, it must be a list enclosed in inverted commas and in lower case, separated by a blank space. If it is not specified, it should be understood that only one value has been specified. This specified value is the "Date" attribute of the request header.</p> <p>The order of the attributes is important and must be the same as the order specified in the list of HTTP headers specified in this field.</p>	<p>The mandatory fields to sign are:</p> <ul style="list-style-type: none"> • digest • x-request-id <p>Conditionally, if they travel and are supported, it can include:</p> <ul style="list-style-type: none"> • psu-id • psu-corporate-id • tpp-redirect-uri
Signature	String	MA	<p>The "signature" parameter must be in Base64 ACCORDING to RFC 4648.</p> <p>The TPP uses the algorithm and the parameters of the header to be signed to form the string to be signed. The chain to be signed is signed with the keyId and the corresponding algorithm. The content must be in Base64.</p>	<p>There are no additional requirements.</p>

9.1.3 Example

We are going to make a host-to-host request with the following text:

```
{
  "instructedAmount" : {
    "currency" : "EUR",
    "amount" : "16.00"
  },
  "debtorAccount" : {
    "iban" : "ES5140000001050000000001",
    "currency" : "EUR"
  },
  "creditorName" : "Cred. Name",
  "creditorAccount" : {
    "iban" : "ES6621000418401234567891",
    "currency" : "EUR"
  },
  "creditorAddress" : {
    "street" : "Ejemplo de calle",
    "buildingNumber" : "15",
    "city" : "Cordoba",
    "postalCode" : "14100",
    "country" : "ES"
  },
  "remittanceInformationUnstructured": "Payment",
  "chargeBearer": "CRED"
}
```

And we also want to add the following headers

- X-Request-ID=a13cbf11-b053-4908-bd06-517dfa3a1861

The operations that we must carry out are the following.

9.1.3.1 Generation of the header "Digest"

To do this, we must hash the body of the message to be sent. It is vital to do this on the final, already serialised content, as subsequent serialisation processes could introduce modifications to the body of the message finally sent, rendering the signature invalid.

It is possible to use the SHA-256 and SHA-512 algorithms following RFC 5843. In our example we will use SHA-256 on the body of the message, obtaining the following result:

- Hexadecimal:
A5F1CF405B28E44ED29507E0F64495859BA877893D2A714512D16CE3BD8
BE562
- Base64: pfHPQFso5E7SIQfg9kSVhZuod4k9KnFFEtFs472L5WI=

Therefore, the value of the "Digest" header that we are going to generate will be:

SHA256=pfHPQFso5E7SIQfg9kSVhZuod4k9KnFFEtFs472L5WI=

The headers that we have so far are:

X-Request-ID=a13cbf11-b053-4908-bd06-517dfa3a1861f]

Digest=SHA256=pfHPQFso5E7SIQfg9kSVhZuod4k9KnFFEtFs472L5WI=

9.1.3.2 Generation of the header "Signature"

The "Signature" header is multivalued, that is, it contains several pairs of sub-headers of the attribute-value type.

Setting the "keyId" value

This field is obtained from the serial number of the certificate in hexadecimal and the DN of the certifying authority that generated the certificate.

In our example we get the following result:

keyId="SN=-5d803f65,CA=CN=REDSYS-AC-EIDAST-C1,OU=PKI,O=REDSYS,C=ES"

Setting the "headers" attribute

It should be noted that this attribute and some others are shown in the Berlin Group document with the first character in uppercase, but in the RFC on which the entity is based its content is always set in lowercase, so we assume that it is a typo.

Here are established the fields that will be considered when making the signature.

headers="digest x-request-id"

Setting the "algorithm" attribute

algorithm = "SHA-256"

Construction of the chain to be signed

The chain that we have left to sign according to point 2.2.3 is the following:

Digest: SHA256=pfHPQFso5E7SIQfg9kSVhZuod4k9KnFFEtFs472L5WI=

X-Request-ID: a13cbf11-b053-4908-bd06-517dfa3a1861f

Signature generation

We sign the chain obtained in the previous point with the private key of our certificate and pass the result to Base64, obtaining in our particular case the following result:

la8LV3Fny2so4c40OkYFtZvr1mOkOVY1n87iKfIggEkXQjZNcyjp9fFkNtQc+5ZVNESdiq
KG8xrawYa5gAm46CvcKChNTPaakiEJHcXM5RZPWN0Ns5HjV5mUY2QzD+g5mwqcW
vXtBr1vg0bZKN8Zt3+uJMN37NQg9tJNE2yKIJIEPIAYOjC2PA/yzGSLOdADnXQut9yRvx
w8gMCjDtRaKdYWmwG6/crX293hGvBUeff1xvTluWhQzyfx4J6WG0v1ZmpnWdZ1LF6
8sToeDGTdu65aVKV2q6qcZzcm5aPV6+mVHX+21Vr6acxiLZdeYUHYJHrzErUN3KJrmt
3w2AL7Dw==

9.1.3.3 Generation of the header "TPP-Signature-Certificate"

This header contains the certificate that we have used in Base64. For reasons of space, only one part is set in the example:

TPP-Signature-Certificate="MIIEWTCCA0GgAwIBAgI....

9.1.3.4 Definitive headers to send

As seen in the previous points, the headers that we must send in the request are:

X-Request-ID=a13cbf11-b053-4908-bd06-517dfa3a1861f

Digest=SHA256=pfHPQFso5E7SIQfg9kSVhZuod4k9KnFFEtFs472L5WI=

Signature=keyId="SN=-5d803f65,CA=CN=REDSYS-AC-EIDAST-
C1,OU=PKI,O=REDSYS,C=ES",algorithm="SHA-256",headers="digest x-request-
id",signature="

la8LV3Fny2so4c40OkYFtZvr1mOkOVY1n87iKfIggEkXQjZNcyjp9fFkNtQc+5ZVNESdiq
KG8xrawYa5gAm46CvcKChNTPaakiEJHcXM5RZPWN0Ns5HjV5mUY2QzD+g5mwqcW
vXtBr1vg0bZKN8Zt3+uJMN37NQg9tJNE2yKIJIEPIAYOjC2PA/yzGSLOdADnXQut9yRvx
w8gMCjDtRaKdYWmwG6/crX293hGvBUeff1xvTluWhQzyfx4J6WG0v1ZmpnWdZ1LF6
8sToeDGTdu65aVKV2q6qcZzcm5aPV6+mVHX+21Vr6acxiLZdeYUHYJHrzErUN3KJrmt
3w2AL7Dw=="

TPP-Signature-Certificate=MIIEWTCCA0GgAwIBAgIEon/...

9.2 HTTP response codes

The HTTP codes followed by this specification and their uses are as follows:

HTTP code	Description
200 OK	<ul style="list-style-type: none"> Response code for PUT and GET requests This code is allowed if the request was repeated due to a time-out. The response can be a 200 or 201 depending on the implementation of the ASPSP The FCS POST request also allows to return a 200 since no new resource is created. Response code for DELETE requests when the request has been made correctly and authorization is not required.
201 Created	Response code for POST requests where a new resource has been created successfully.
202 Accepted	Response code for DELETE requests when a payment resource can be canceled but requires authorization of the cancellation by the PSU.
204 No Content	<p>Response code for DELETE requests where the consent resource has been successfully deleted. The code indicates that the response was made, but no content is returned.</p> <p>Also used in DELETE requests of a payment start where authentication is not necessary.</p>
400 Bad Request	A validation error occurred. This code covers syntax errors in requests or incorrect data in the payload.
401 unauthorized	The TPP or the PSU are not properly authorized to make the request. Retry the request with correct authentication information.
403 Forbidden	Returned if the resource that was referenced in the path exists but cannot be accessed by the TPP or the PSU. This code should only be used for non-sensitive identifiers as this could reveal that the resource exists but cannot be accessed.
404 Not Found	<p>Returned if the resource that was referenced in the path exists but cannot be accessed by the TPP or the PSU.</p> <p>When in doubt if a specific path id is sensitive or not, use this code instead of 403.</p>
405 Method Not Allowed	This code is sent only when the method (POST, PUT, GET ...) is not supported on a specific endpoint.

	Response code for DELETE in case of payment cancellation, where a payment start cannot be canceled due to legal or other operational reasons.
406 Not Acceptable	The ASPSP cannot generate the content that the TPP specifies in the Accept header field
408 Request Timeout	The server is still working correctly, but the request has timed out.
409 Conflict	The request could not be completed due to a conflict with the current state of the referenced resource.
415 Unsupported Media Type	The TPP has requested a "media type" that the ASPSP does not support.
429 Too Many Requests	The TPP has exceeded the maximum number of requests allowed by consent or by the RTS
500 Internal Server Error	Internal server error has occurred.
503 Service Unavailable	The ASPSP server is currently unavailable. It is generally a temporary state.

9.3 Return Codes

Allowed return codes and associated HTTP response codes.

	HTTP code	Code	Description
SIGNATURE CERTIFICATE	401	CERTIFICATE_INVALID	The content of the signing certificate is invalid.
	401	ROLE_INVALID	The TPP does not have the correct PSD2 roles to access the service.
	401	CERTIFICATE_EXPIRED	The signature certificate has expired.
	401	CERTIFICATE_BLOCKED	The signature certificate has been blocked by the ASPSP.
	401	CERTIFICATE_REVOKED	The signature certificate has been revoked by the QTSP.

PSD2 - APIs Implementation Guide v1.1 for TPPs

	401	CERTIFICATE_MISSING	The signature certificate was not present in the request.
SIGNATURE	401	SIGNATURE_INVALID	The signature is not correct.
	401	SIGNATURE_MISSING	The signature is not included in the message being mandatory.
GENERAL	400	FORMAT_ERROR	<p>The format of certain fields in the request is wrong. The fields will be indicated.</p> <p>This applies to the body and header fields. It also applies in cases where these entries refer to non-existent or erroneous data instances.</p>
	400	PARAMETER_NOT_CONSISTENT	<p>Parameters sent by the TPP are not consistent.</p> <p>It only applies to query parameters.</p>
	400	PARAMETER_NOT_SUPPORTED	The parameter is not supported by ASPSP. It will only be used in those parameters whose support is optional for ASPSP.
	401	PSU_CREDENTIALS_INVALID	The PSU-ID is not related to the ASPSP or is locked, or the password or the OTP was incorrect.
	400 (payload) 405 (HTTP method)	SERVICE_INVALID	The requested service is not valid for the indicated resource, or the data sent.
	403	SERVICE_BLOCKED	The service is not available to the PSU due to a channel blocking by the ASPSP.
	401	CORPORATE_ID_INVALID	The PSU-Corporate-ID has failed to link on the ASPSP systems.

PSD2 - APIs Implementation Guide v1.1 for TPPs

	403 (if resource in path) 400 (if resource in header)	CONSENT_UNKNOWN	The Consent-ID does not match for the requested TPP and ASPSP.
	401	CONSENT_INVALID	The consent was created by the TPP, but it is not valid for the requested resource / service. Or the definition of consent is incomplete or invalid.
	401	CONSENT_EXPIRED	The consent was created by the TPP, but it has expired and needs to be renewed.
	401	TOKEN_UNKNOWN	The received token is unknown to the TPP.
	401	TOKEN_INVALID	The token is associated with the TPP, but it is not valid for the service / resource you are trying to access.
	401	TOKEN_EXPIRED	The token is associated with the TPP, but it has expired and needs to be renewed.
	404 (if account-id in path) 403 (if other resource in path) 400 (if it goes on payload)	RESOURCE_UNKNOWN	The requested resource is unknown to the TPP.
	403 (if resource in path) 400 (if resource in payload)	RESOURCE_EXPIRED	The requested resource is associated with the TPP, but it has expired and will no longer be available.

PSD2 - APIs Implementation Guide v1.1 for TPPs

	400	RESOURCE_BLOCKED	The directed resource is not routable by the request. This can be blocked, for example, by a grouping in the "signing basket".
	400	TIMESTAMP_INVALID	Timestamp not in accepted time period.
	400	PERIOD_INVALID	Requested time period out of range.
	400	SCA_METHOD_UNKNOWN	The SCA method selected in the authentication method selection request is unknown or cannot be related by the ASPSP to the PSU.
	400	SCA_INVALID	The HTTP method used on the authorization resource is blocked because the resource state equals "failed".
	409	STATUS_INVALID	The addressed resource does not allow additional authorization.
OAuth2	302	invalid_request	The request is not well formed due to missing parameters, unsupported value or repeated parameters.
	302	unauthorized_client	The authenticated client is not authorized to use this type of authorization.
	302	access_denied	The resource owner or authorization server denies the request.
	302	unsupported_response_type	The authorization server does not support the method used to obtain the authorization code.
	302	invalid_scope	The requested scope is invalid, unknown, or wrongly formed.

PSD2 - APIs Implementation Guide v1.1 for TPPs

	302	server_error	Error 500 that cannot be returned in a redirect. It is returned with this code.
	302	temporarily_unavailable	The authorization server is unable to process the request momentarily, due to temporary overload or maintenance.
	400	invalid_request	The request is not well-formed because of missing parameters, unsupported value, repeated parameters, includes multiple credentials or uses more than one client authentication mechanism.
	401	invalid_client	Client authentication failed
	400	invalid_grant	The provided authorization or refresh token is invalid, expired, revoked, redirect URI mismatch, or was issued to another client.
	400	unauthorized_client	The authenticated client is not authorized to use this type of authorization.
	400	unsupported_grant_type	The requested authorization type is not supported by the authorization server.
	400	invalid_scope	The requested scope is invalid, unknown, malformed, or exceeds what is allowed.
PIS	403	PRODUCT_INVALID	The requested paid product is not available for the USP.
	404	PRODUCT_UNKNOWN	The requested payment product is not supported by the ASPSP
	400	PAYMENT_FAILED	Payment failed. For example, for risk management reasons.

	400	EXECUTION_DATE_INVALID	The requested run date is not a valid run date for the ASPSP.
	405	CANCELLATION_INVALID	The directed payment is not cancellable. For example, a long time or legal restrictions have passed.
AIS	401	CONSENT_INVALID	The consent was created by the TPP, but it is not valid for the requested resource / service. Or the definition of consent is incomplete or invalid.
	400	SESSIONS_NOT_SUPPORTED	The combined service indicator is not supported by the ASPSP to which the request is directed.
	429	ACCESS_EXCEEDED	Account accesses have exceeded the accesses allowed per day with no PSU present.
	406	REQUESTED_FORMATS_INVALID	The format requested in the Accept field does not correspond to those offered by the ASPSP.
FCS	400	CARD_INVALID	Card numbering is unknown to ASPSP or not associated with PSU.
	400	NO_PIIS_ACTIVATION	The PSU has not activated the account for use by the PIIS associated with the TPP.

9.4 Transaction states

Code	Name	Description
ACCP	AcceptedCustomerProfile	The prior verification of the technical validation was correct. The client profile check was also successful.

ACSC	AcceptedSettlementCompleted	<p>The settlement in the issuer's account has been completed.</p> <p>Usage: it is used by the first agent (the issuer's ASPSP through the HUB) to inform the issuer that the transaction has been completed.</p> <p>Important: The reason for this state is to provide the state of the transaction, not for financial information. It can only be used after a bilateral agreement.</p>
RCVD	Received	The payment initiation has been received by the agent (the ASPSP through the HUB)
PATC	PartiallyAcceptedTechnicallyCorrect	<p>Payment starts that have been authorized by at least one USP, but have not yet been finally authorized by all applicable USPs. (Multilevel SCA)</p> <p>Note: needs ISO 20022 approval</p>
RJCT	Rejected	The payment initiation or the individual transaction included in the payment start has been rejected.
CANC	Cancelled	<p>The start of payment has been canceled before its execution.</p> <p>Note: needs ISO 20022 approval</p>

9.5 Consent states

Code	Description
received	The consent has been received and is technically correct. The data has not been authorized yet.
rejected	The consent has been refused.
valid	The consent is accepted and valid to make requests to read the data and specified in the consent.
revokedByPsu	The consent has been revoked by the PSU towards the ASPSP.
expired	The consent has expired.
terminatedByTpp	The corresponding TPP has terminated the consent using the DELETE request on the consent resource created.

9.6 Types of balances

Code	Description
closingBooked	Account balance at the end of the pre-agreed period for the report. It is the sum of the balances "openingBooked" at the beginning of the period and all entries noted in the account during the pre-agreed period for the report.
openingBooked	Account balance at the beginning of the reporting period. It is always equal to the "closingBooked" balance of the previous period's report.

9.7 Types of commission sharing

Code	Description
DEBT	All transaction charges are paid by the payer
CRED	All transaction charges are paid by the beneficiary
SHAR	Shared charges. Issuer and beneficiary bear the corresponding charges on their side.
SLEV	The charges to be applied follow the rules agreed at the level of service and / or scheme

9.8 SCA states

Code	Description
received	The authorization resource has been created successfully.
scaMethodSelected	The PSU / TPP has selected the SCA flow. If the SCA method is chosen implicitly because there is only one SCA method available, then this state is the first state to report instead of "received"
started	The SCA flow has been started.
finalised	The SCA flow has been completed successfully.
failed	SCA flow has failed.

FAQ: Frequently Asked Questions

1. Which types of payment are available and supported?

The following table shows the products and types of payments supported:

		National	International (cross-border)	
		SEPA	SEPA	NO SEPA
Payment Product	Punctual	Yes	Yes	Yes
	Deferred	Yes	Yes	Yes
	Periodic	Yes	No	No
	Immediate (TARGET2)	Yes	No	No

2. What types of accounts are supported?

Retail and business/corporate users.

Corporate users, in order to have access to the PSD2 channel, must be authorised by the administrator of their company, who must assign them the necessary permissions.

3. How many days of transactions can be recovered after the creation of the valid consent?

With a valid consent, you can recover up to 2 years of transactions from the date of application.

4. What are the supported transaction statuses?

The supported values are "booked", "pending", and either "both". In addition, with the new version /v1.1/ "information" is also supported for retrieving standing orders.

5. Can the PSU revoke a consent via its online banking portal?

PSD2 consents can only be revoked via the APIs, using the consent removal endpoint provided.

6. How long does the PSU have to request transactions older than 90 days?

The customer has 20 minutes from the creation of a consent to request transactions older than 90 days.

7. How many transactions come per page?

A total of 20 transactions are listed per page.

8. In what order are the transactions listed?

Transactions are returned in descending order by date. If a new transaction is made, it will be listed at the top.

9. Is it allowed to create and renew several tokens simultaneously for the same PSU?

No. A customer may only use one access token at a time. Thus, once a customer has exchanged a code for an access token, and the latter is valid, he could operate without any inconvenience. In other words, the old access token is invalidated when a new one is created.

Consequently, there is no differentiation between tokens for AIS or PIS requests.