renta4banco

PSD2-TPP Technical Design

July 2022

Version: 1.8.5

Authorization 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: - Chapter 2.0: supported services review - Chapter 4: general review

TABLE OF CONTENTS

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

	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.

2. SYSTEM OVERVIEW

The following shows the different Figure 1: Core Module Diagram Functional Modules of which it is composed, and which will be detailed later on.

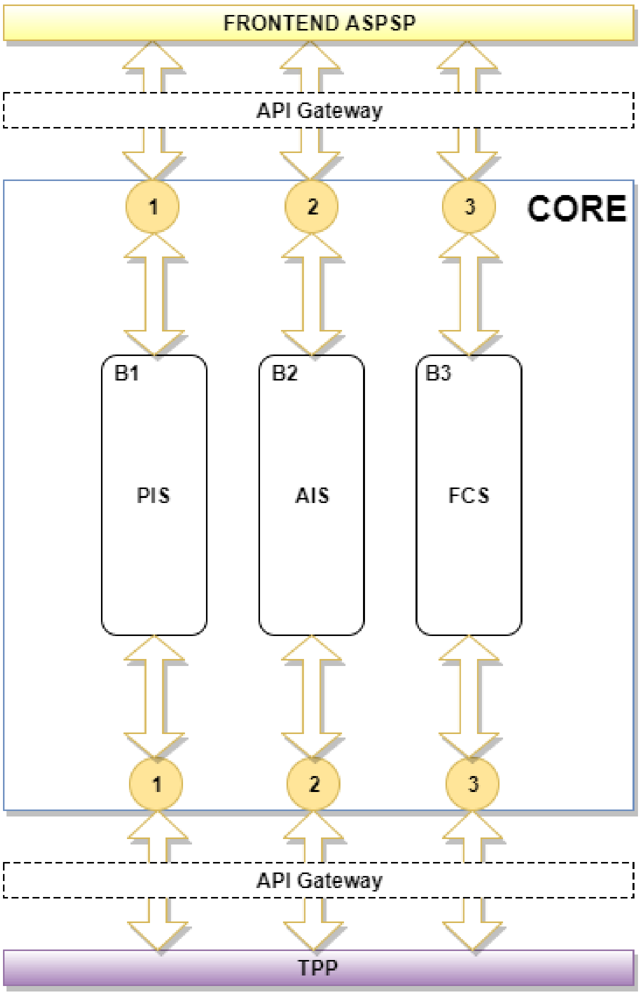


Figure 1: Core Module Diagram

Service		Function	State
CORE	PIS	Start of simple single-signature payment	Available
		Start of recurring payments	Available
		Initiation of multiple payments/bulk	Available
		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 and / or cards	Available
		Retrieve consent 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	Available
		Reading card account details	Available
		Reading of card account balances	Available
		Reading card account transactions	Available
	FCS	Establish consent	Available
		Retrieve consent information	Available
		Check consent state	Available
		Remove consent	Available
		Confirmation of funds	Available
	SCA	SCA per flow redirect	Available
		Uncoupled flow SCA	Available

		SCA embedded	Not Supported
	Common processes	Start explicit authorization	Available
		SCA state query	Available
		Get authorization sub-resources	Available
		Update authorization data	Available
	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
	AIS	Notice of data available in PUSH mode	GN pending
	DIR. TPPs	List of available TPPs	Available
		TPP information query	Available
		New TPP notification	Available

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	Mandat.	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=NzdmZjA4YjY5M2M2NDYyMmVjOWFmMGNmYTZiNTU3MjVmNDI4NTRlMzJkYzE3ZmNmMDE3ZGFmMjhhNTc5OTU3OQ==
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.	String	MA	^.{1,5000}$

	See 9.1Signature for more information.			EX: TPP-Signature-Certificate: MIIHgzCCBmugAwIBAgII ZzZvBQlt0UcwDQYJ…… …….KoZIhvcNAQELBQA wSTELMAkGA1UEBhMC VVMxEzARBgNVBA

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)

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	Mandat.	Format
PSU-IP-Address	IP address of the HTPP 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
PSU-Accept-Language	Accept language header of the HTTP request between PSU and the TPP.	String	OP	^.{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: • 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** ^[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** ^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	OB	Redirection to the ASPSP login website to obtain the authCode.
/token	POST	OB	Allows to send the authCode to obtain the access token.
/token	POST	OB	Refresh the access token if it has expired.

5.2 Payment Endpoints

Endpoint	Method	Cond.	Description
/payments/{payment-product}	POST	OB	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	OB	Gets the details of an initiated payment.
/payments/{payment-product}/{paymentId}/state	GET	OB	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	OB	Gets the details of an initiated standing order for periodic/recurrent payment.
/periodic-payments/{payment-product}/{paymentId}/state	GET	OB	Gets the state of the standing order transaction for periodic/recurrent payment.
{payment-service}/{payment-product}/{paymentId}/authorisations	POST	OB	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	OB	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}/authorisations/{authorisationId}	PUT	MA	Updates data in the authorization resource, if necessary.
{payment-service}/{payment-product}/{paymentId}	DELETE	OP	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. The response to this DELETE command will tell the TPP when: • The access method was rejected • The access method was correct • The access method is generally applicable but requires an additional authorization process.
{payment-service}/{payment-product}/{paymentId}/cancellation-authorisations	POST	OP	Initiates the authorisation of the cancellation of the payment accessible under the paymentId resource if requested by the ASPSP (e.g. DELETE method is not sufficient) and if applicable for the payment service, and received within the time period that is cancellable.
{payment-service}/{payment-product}/{paymentId}/ cancellation-authorisations	GET	MA	Gets the list of cancellation authorization sub-resources that have been created.

			Note: if the POST command on this endpoint is supported, then this GET method must also be supported.
{payment-service}/{payment-product}/{paymentId}/ cancellation-authorisations/{authorisationId}	GET	MA	Gets the SCA state of the cancellation authorisation. **Note**: if the POST command on this endpoint is supported, then this GET method must also be supported.
{payment-service}/{payment-product}/{paymentId}/ cancellation-authorisations/{authorisationId}	PUT	MA	Updates data in the authorization resource, if necessary. **Note**: if the POST command on this endpoint is supported, then this PUT method must also be supported.

5.3 AccountEndpoints

Endpoint	Method	Cond.	Description
/accounts	GET	MA	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. **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.

/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 from, and date to.
/accounts/{account-id}/transactions?withBalance	GET	MA	Obtains a list of movements (transactions) of the accessed account together with balances.

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 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. 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. As a last option, the ASPSP can accept requests with the following access permissions: • 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}/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 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.
/consents/{consentId}/authorisations/{authorisationId}	PUT	MA	Updates data in the authorization resource, if necessary.

5.5 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.
/consents/confirmation-of-funds/{consentId}/authorisations/{authorisationId}	PUT	MA	Updates data in the authorization resource, if necessary.

5.6 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.

5.7 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.
/tpps	GET	OP	Get the list of TPPs in the Hub.
/tpps/{tppId}	GET	OP	Get the detail of a TPP.

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 2: OAuth2 scenario as a pre-step is represented with its "Authorisation Code Grant" flow and the involvement of the parts.

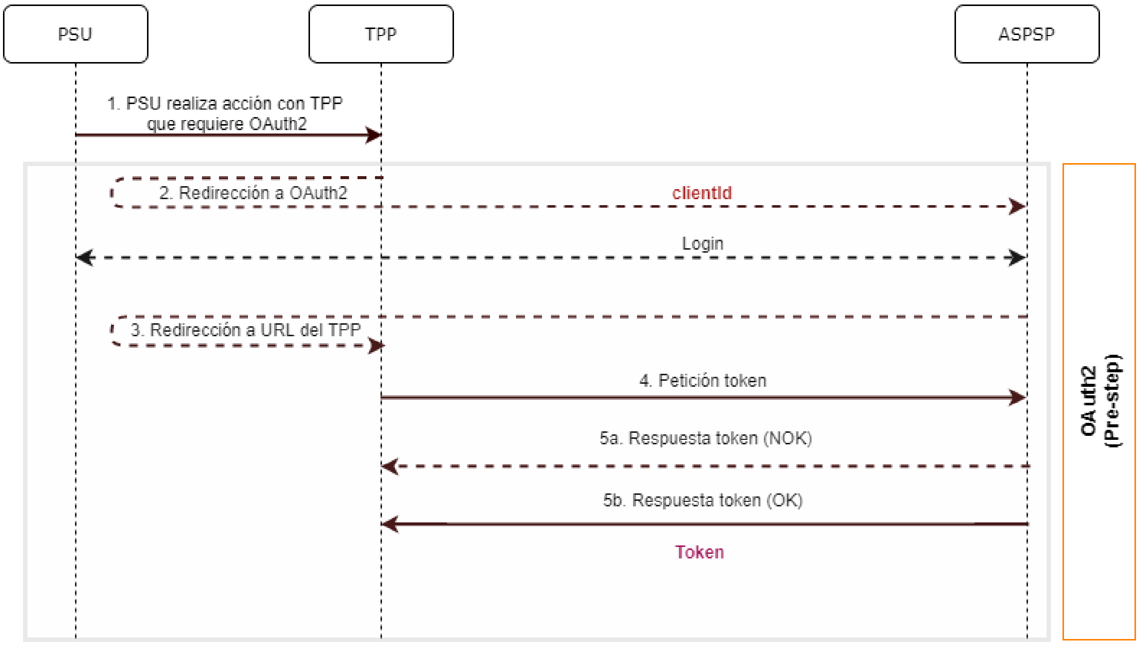


Figure 2: OAuth2 scenario as a pre-step

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%2Ftpp%2Eexample%2Ecom%2Fcb&code_challeng
e=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=SplxlOBeZQQYbYS6WxSbIA&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.

```
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": "G5Qx2TlKWIAtGzv3JOkF0X"
}
```

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}

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

GET /{aspsp}/biometric/app-to-app/business/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	Mandat.	Format
response_type	The value must be set to "code".	String	MA	Ex: response_type = code
client_id	“organizationIdentifier” provided in the eIDAS certificate formed as: - 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 This registration number will be that of the HUB or the TPP depending on the ASPSP configuration.	String	MA	^.{1,70}$ Ex: client_id=PSDES-RDS-4000
scope	Scope possible: - PIS - AIS - SVA You can specify more than one by separating it by a space (% 20).	String	MA	^.{1,64}$ Ex: scope=PIS%20AIS%20SVA
state	Opaque value generated by the TPP. Used to prevent cross-site request forgery XSRF attacks.	String	MA	^.{1,64}$ Ex: state = XYZ
redirect_uri	URL back to the HUB where the authorization code "code" that will be used later to obtain the	String	MA	^.{1,250}$ Ex:

	access token will be reported.			redirect_uri=https%3A%2F%2Fwww%2Ehub%2Ecom%2Fcb
code_challenge	PKCE challenge used to prevent code injection attacks. According to RFC 7636.	String	MA	^.{1,128}$ Ex: code_challenge=E9Melhoa2OwvFrEMTJguCHaoeK1t8URWbuGJSstw-cM
code_challenge_method	Method to verify the code that can be "plain" or "S256". Preferred S256 (SHA 256)	String	OP	^.{1,120}$ Ex: code_challenge_method = S256
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	^.{1,70}$ Ex: second_client_id=PSDES-BDE-3DFD246
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: • staff • business	String	OP	Ex: app_to_app_preferred=true

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 life time of no more than 10 minutes.	String	MA	[A-Za-z0-9]{32} Ex: code=SplxlOBeZQQYbYS6WxSbIA
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	Mandat.	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=SplxlOBeZQQYbYS6WxSbIA&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	Mandat.	Format
provider	URL of the HUB where the service is released.	String	MA	Ex: hub.example.es

Request Parameters

Field	Description	Type	Mandat.	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: - 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	String	MA	^.{1,70}$ Ex: client_id=PSDES-RDS-4000
code	Authorisation code returned by the ASPSP in the previous request for an authorisation code application	String	MA	^.{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	^.{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**	**Mandat.**	**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	Mandat.	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=SplxlOBeZQQYbYS6WxSbIA&redirect_uri=https%3A%2F%2Fwww%2Etpp%2Ecom%2Fcb&code_verifier=dBjftJeZ4CVP-mB92K27uhbUJU1p1r_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": "2YotnFZFEjr1zCsicMWpAA",
        "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
{
        Ex: "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 3: 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.

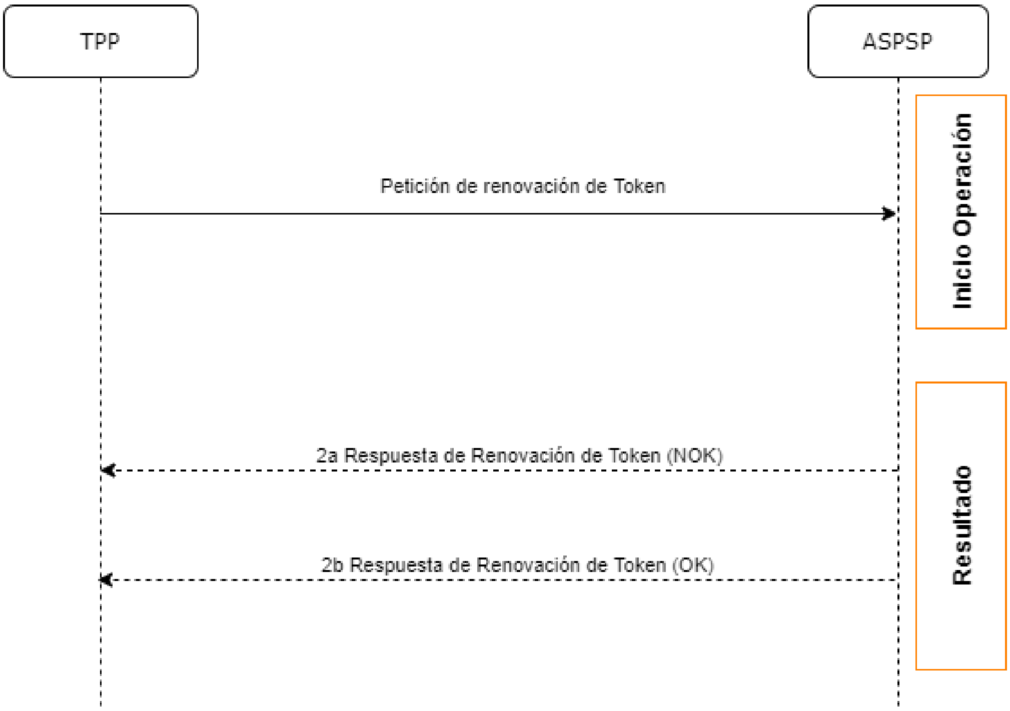

TPP
ASPSP
Petición de renovación de Token
Inicio Operación
2a Respuesta de Renovación de Token (NOK)
2b Respuesta de Renovación de Token (OK)
Resultado

Figure 3: 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=tGzv3JOkF0XG5Qx2TlKWIA
```

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	“organizationIdentifier” provided in the eIDAS certificate formed as: - 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	String	MA	^.{1,70}$ Ex: client_id=PSDES-RDS-4000
refresh_token	Refresh token to obtain an unexpired accessToken.	String	MA	^.{1,64}$ Ex: refresh_token=tGzv3JOkF0XG5Qx2TlKWIA

Header

No additional data is specified.

Body

No additional data is specified.

6.2.3 Response

Field	Description	Type	Mandat.	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": "28JD3JOkF0NM5Qx2TlCCC"

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": "83kdFZFEjr1zCsicMWBB",
        "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 4: Start of payment with OAuth2 as pre-step and SCA flow by redirectionthe 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)
- TPP-Explicit-Authorization-Preferred: false - TPP preference to initiate the payment authorization process implicitly
- The PSU has only one SCA method

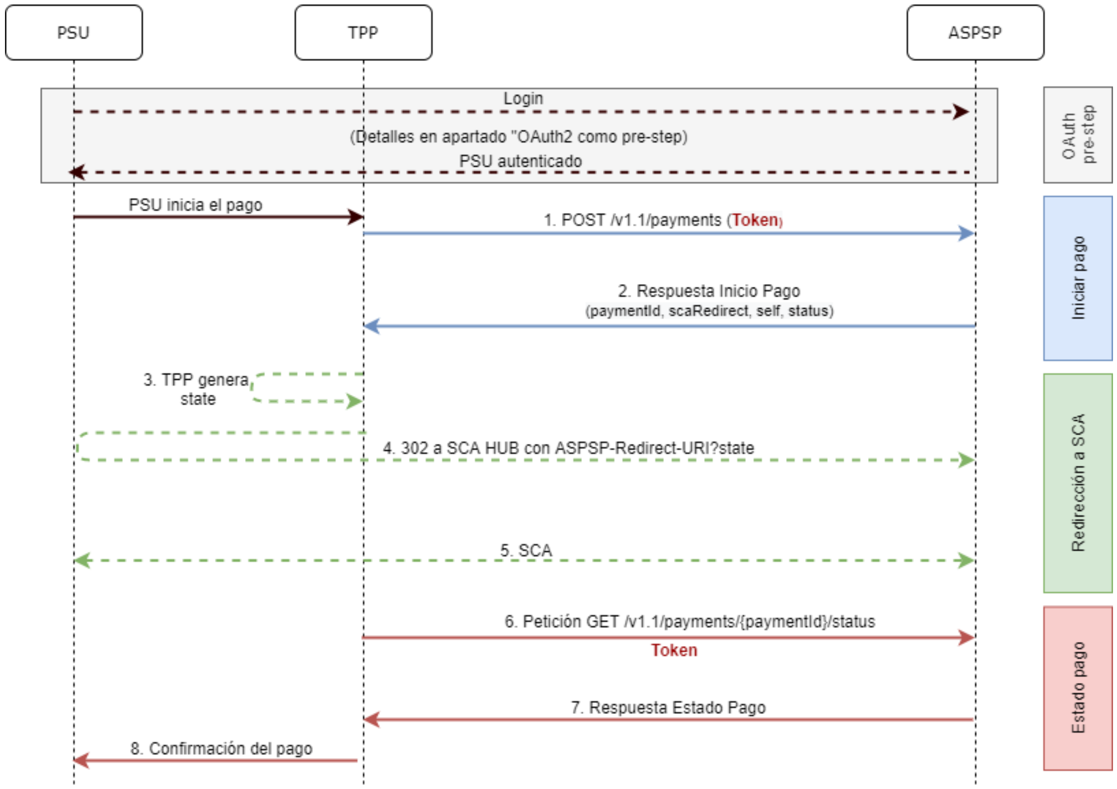

PSU
TPP
ASPSP
Login
(Detalles en apartado "OAuth2 como pre-step)
PSU autenticado
OAuth pre-step
PSU inicia el pago
1. POST /v1.1/payments (Token)
Iniciar pago
2. Respuesta Inicio Pago
(paymentId, scaRedirect, self, status)
3. TPP genera state
4. 302 a SCA HUB con ASPSP-Redirect-URI?state
Redirección a SCA
5. SCA
6. Petición GET /v1.1/payments/{paymentId}/status
Token
Estado pago
7. Respuesta Estado Pago
8. Confirmación del pago

Figure 4: 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 4: 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: implicit 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 methodthe 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)
- 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.

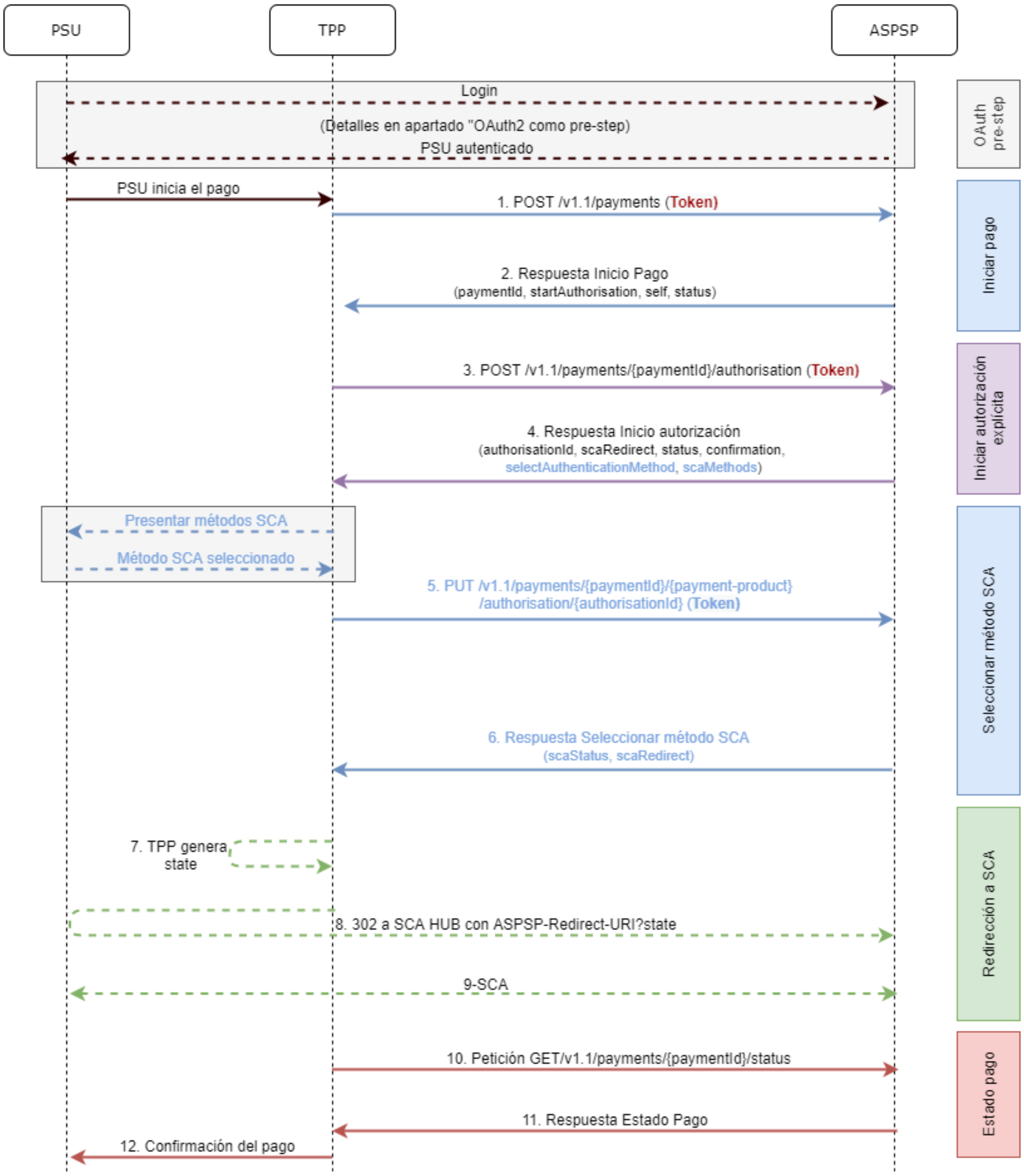

PSU
TPP
ASPSP
Login
(Detalles en apartado "OAuth2 como pre-step)
PSU autenticado
OAuth pre-step
PSU inicia el pago
1. POST /v1.1/payments (Token)
2. Respuesta Inicio Pago
(paymentId, startAuthorisation, self, status)
Iniciar pago
3. POST /v1.1/payments/{paymentId}/authorisation (Token)
4. Respuesta Inicio autorización
(authorisationId, scaRedirect, status, confirmation,
selectAuthenticationMethod, scaMethods)
Iniciar autorización explícita
Presentar métodos SCA
Método SCA seleccionado
5. PUT /v1.1/payments/{paymentId}/{payment-product}
/authorisation/{authorisationId} (Token)
6. Respuesta Seleccionar método SCA
(scaStatus, scaRedirect)
Seleccionar método SCA
7. TPP genera state
8. 302 a SCA HUB con ASPSP-Redirect-URI?state
9-SCA
Redirección a SCA
10. Petición GET/v1.1/payments/{paymentId}/status
11. Respuesta Estado Pago
12. Confirmación del pago
Estado pago

Figure 5: 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: • 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
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** ^[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	This field will be ignored by the ASPSP. The session support is specified by the access token.	String	OP	^.{1,36}$ 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
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 HTPP 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.	Boolean	OP	Ex: TPP-Redirect-Preferred: true

	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**			
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.	Boolean	OP	Ex: TPP-Explicit-Authorisation-Preferred: false

	Note: ASPSP might not take it into account if it doesn't support it.			
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
TPP-Rejection-NoFunds-Preferred	**Note:** This field will be ignored if it is reported by the TPP.	String	OP	
TPP-Notification-URI	**Note:** This field will be ignored if it is reported by the TPP.	String	OP	
TPP-Notification-Contained-Preferred	**Note:** This field will be ignored if it is reported by the TPP.	String	OP	

Body

The content of the Body is defined in 8.16 SinglePayment following the conditions of the following table.

The fields marked as mandatory (OB) and optional (OP) are supported by the ASPSP with this type of condition.

The fields marked as COND depend on each ASPSP.

Field	SCT	SCT INST	Target 2	Cross Border CT
EndToEndIdentification*	NA	NA	NA	NA
instructionIdentification	COND	COND	COND	COND
debtorName	COND	COND	COND	COND
debtorAccount	MA	MA	MA	MA
debtorId	COND	COND	COND	COND
ultimateDebtor	COND	COND	COND	COND
instructedAmount	MA	MA	MA	MA

currencyOfTransfer	COND	COND	COND	COND
exchangeRateInformation	COND	COND	COND	COND
creditorAccount	MA	MA	MA	MA
creditorAgent	OP	OP	OP	OB/OP
creditorAgentName	COND	COND	COND	COND
CreditorName	MA	MA	MA	MA
creditorId	COND	COND	COND	COND
creditorAddress	OP	OP	OP	OP
creditorNameAndAddress	COND	COND	COND	COND
ultimateCreditor	COND	COND	COND	COND
purposeCode	COND	COND	COND	COND
chargeBearer	COND	COND	COND	COND
serviceLevel	COND	COND	COND	COND
remittanceInformationUnstructured	OP	OP	OP	OP
remittanceInformationUnstructuredArray	COND	COND	COND	COND
remittanceInformationStructured	COND	COND	COND	COND
remittanceInformationStructuredArray	COND	COND	COND	COND
requestedExecutionDate	n.a.	n.a.	n.a.	n.a.
requestedExecutionTime	n.a.	n.a.	n.a.	n.a.

***NOTE: If you want to send the endToEndId field, you must report it in the body remittanceInformationUnstructured field. The best practices guide provides how to send the endToEndId field within that field.**

6.3.2.2 Response

Header

Field	Description	Type	Mandat.	Format
Location	Contains the link to the generated resource.	String	OB	^.{1,512}$ Ex: Location: /v1.1/payments/{payment-product}/{payment-id}

X-Request-ID	Unique identifier of the operation assigned by the TPP.	String	OB	**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: • EMBEDDED • DECOUPLED • REDIRECT The OAuth based SCA will be taken as REDIRECT.	String	COND	Ex: ASPSP-SCA-Approach: REDIRECT
ASPSP-Notification-Support	Not in use. Resource state notification services not supported	Boolean	NA	
ASPSP-Notification-Content	Not in use. Resource state notification services not supported	String	NA	

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"
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. If equal to "false", the transaction will not imply any additional commission for the PSU.	Boolean	OP	Ex: "transactionFeeIndicator": true
currencyConversionFee	It could be used by the ASPSP to carry currency-specific conversion fees associated with the initiated credit transfer.	Amount	OP	Ex: "currencyConversionFee": {…}
estimatedTotalAmount	Amount which is estimated to be withdrawn from the issuer's account. Note: this amount includes commissions.	Amount	OP	Ex: "estimatedTotalAmount": {…}
estimatedInterbankSettlementAmount	Estimated amount to be transferred to the beneficiary.	Amount	OP	Ex: "estimatedInterbankSettlementAmount": {…}
scaMethods	This element is contained if SCA is required and if the PSU can choose between different authentication methods. If this data is contained, the link "startAuthorisationWithAuthenticationMethodSelection" will also be reported. These methods must be presented to the PSU. **Note:** Only if ASPSP supports SCA method selection	List<AuthenticationObject>	COND	Ex: "scaMethods": […]
chosenScaMethod	**NOT SUPPORTED IN THIS VERSION. ONLY EMBEDDED**	AuthenticationObject	COND	

_links	List of hyperlinks to be recognized by the TPP. Supported types in this response: • 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 contained under the same conditions as the "scaMethods" field • 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-	Links	OB	Ex: "_links": {…}

	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": […]

6.3.2.3 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 2YotnFZFEjr1zCsicMWpAA

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"
        },
        "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"
                }
        }
}
```

Request example for decoupled SCA

```
POST https://hub.example.es/asp-name/v1.1/payments/sepa-credit-transfers
Content-Encoding: gzip
Content-Type: application/json
X-Request-ID: 10391c7e-ad88-49ec-a2ad-00aacb1f6541
Authorization: Bearer 2YotnFZFEjr1zCsicMWpAA
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: false
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 decoupled flow with implicitly created authorisation sub-resource

```
HTTP/1.1 201 Created
X-Request-ID: 10391c7e-ad88-49ec-a2ad-00aacb1f6541
ASPSP-SCA-Approach: DECOUPLED
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": {
                "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"
                }
        },
        "psuMessage": "Please use your XXX Bank application to authorize the payment"
}
```

6.3.3 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

- **Daily payments**: the "dayOfExecution" field is not necessary. The first payment is the "startDate" and, from there, the payment is made every day
- **Weekly payments**: if "dayOfExecution" is required, the possible values are from 01 = Monday to 07 = Sunday. If "dayOfExecution" is not required, "startDate" is taken as the day of the week the payment is made. (If "startDate" is Thursday, the payment would be made every Thursday)
- **Bi-weekly payments**: same rule applies as weekly payments.
- **Monthly payments or higher:** possible values range from 01 to 31. Using 31 as the last day of the month

6.3.3.1 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	OB	Ex: hub.example.es
aspsp	Name of the ASPSP to which the request is to be made.	String	OB	Ex: aspsp-name

payment-product	Paid product to use. List of supported products: • sepa-credit-transfers • instant-sepa-credit-transfers • target-2-payments • cross-border-credit-transfers	String	OB	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 8.16 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	OB	**ISODate** xEx: "startDate":"2018-12-20"
executionRule	Supported values: • 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. The ASPSP may reject the request due to the communicated value if the Online Banking rules do not support this execution rule.	String	OP	Ex: "executionRule":"following"

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: • Daily • Weekly • EveryTwoWeeks • Monthly • EveryTwoMonths • Quarterly • Semi Annual • Annual	String	MA	**EventFrequency7Code de ISO 20022** Ex: "frequency": "Monthly"
dayOfExecution	"31" is last. Following the regular expression \d{1,2} The date refers to the ASPSP time zone. Only if supported in ASPSP Online Banking.	String	COND	\d{1,2} Ex: "dayOfExecution": "01"

The fields marked as mandatory (MA) and optional (OP) are supported by the ASPSP with this type of condition.

The fields marked as COND depend on each ASPSP.

Field	SCT	SCT INST	Target 2	Cross Border CT
EndToEndIdentification*	NA	NA	NA	NA
instructionIdentification	COND	COND	COND	COND
debtorName	COND	COND	COND	COND
debtorAccount	MA	MA	MA	MA
debtorId	COND	COND	COND	COND
ultimateDebtor	COND	COND	COND	COND

instructedAmount	MA	MA	MA	MA
currencyOfTransfer	COND	COND	COND	COND
exchangeRateInformation	COND	COND	COND	COND
creditorAccount	MA	MA	MA	MA
creditorAgent	OP	OP	OP	OB/OP
creditorAgentName	COND	COND	COND	COND
CreditorName	MA	MA	MA	MA
creditorId	COND	COND	COND	COND
creditorAddress	OP	OP	OP	OP
creditorNameAndAddress	COND	COND	COND	COND
ultimateCreditor	COND	COND	COND	COND
purposeCode	COND	COND	COND	COND
chargeBearer	COND	COND	COND	COND
serviceLevel	COND	COND	COND	COND
remittanceInformationUnstructured	OP	OP	OP	OP
remittanceInformationUnstructuredArray	COND	COND	COND	COND
remittanceInformationStructured	COND	COND	COND	COND
remittanceInformationStructuredArray	COND	COND	COND	COND
requestedExecutionDate	n.a.	n.a.	n.a.	n.a.
requestedExecutionTime	n.a.	n.a.	n.a.	n.a.

***NOTE: If you want to send the endToEndId field, you must report it in the body remittanceInformationUnstructured field. The best practices guide provides how to send the endToEndId field within that field.**

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

The same as those defined in the section 6.3.2.2

6.3.3.3 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 2YotnFZFEjr1zCsicMWpAA
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"
}
```

6.3.4 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.

6.3.4.1 Request

Endpoint

GET {provider}/{aspsp}/v1.1/{payment-service}/{payment-product}/{paymentId}/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
payment-service	Possible values are: • payments • periodic-payments	String	MA	Ex: {provider} / {aspsp} /v1.1/payments
payment-product	Paid product to use. List of supported products: • 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	OB	^.{1,36}$ Ex: 1234-qwer-5678

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** ^[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: • application/json	String	OP	^.{1,50}$ Ex: Accept: application/json

Body

No additional data is specified.

6.3.4.2 Response

Header

Field	Description	Type	Mandat.	Format
X-Request-ID	Unique identifier of the request assigned by the TPP.	String	OB	**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	Mandat.	Format
transactionStatus	State of the payment transaction. Defined values in 9.4 Transaction states	String	MA	**ISO20022** Ex: "transactionStatus": "ACCP"
fundsAvailable	This data is contained if supported by the ASPSP, if a confirmation of funds has been made and if the "transactionStatus" is any of the following: • ATCT • ACWC • ACCP	Boolean	COND	Ex: "fundsAvailable": true
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": […]

6.3.4.3 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 2YotnFZFEjr1zCsicMWpAA

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
{
        "transactionStatus": " ACCP",
        "fundsAvailable": true
}
```

6.3.5 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.

6.3.5.1 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
aspsp	Name of the ASPSP to which the request is to be made.	String	MA	Ex: aspsp-name
payment-service	Possible values are: • payments • periodic-payments	String	MA	Ex: {provider} / {aspsp} /v1.1/payments

payment-product	Paid product to use. List of supported products: • 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 6.3.4.1

Body

No additional fields are specified.

6.3.5.2 Response

Header

The same as those defined in the section 6.3.4.2

Body

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

- 6.3.2Payment start
- 6.3.3 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<TppMessage>	OP	Ex: "tppMessage": […]

6.3.5.3 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

Authorization: Bearer 2YotnFZFEjr1zCsicMWpAA

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",
}
```

6.3.6 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.

6.3.6.1 Request

Endpoint

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

Path

Field	Description	Type	Mandat.	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: • 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** ^[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
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

Body

No additional data is specified.

6.3.6.2 Response

HTTP Code

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

202 if the DELETE command is not sufficient and authorization from the PSU is required.

Header

The same as those defined in the section 6.3.4.2

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"

scaMethods	This element is contained if SCA is required and if the PSU can choose between different authentication methods. If this data is contained, the link "startAuthorisationWithAuthenticationMethodSelection" will also be reported. These methods must be presented to the PSU. **Note:** Only if ASPSP supports SCA method selection	List<AuthenticationObject>	COND	Ex: "scaMethods": […]
chosenScaMethod	**NOT SUPPORTED IN THIS VERSION.**	AuthenticationObject	COND	
_links	List of hyperlinks to be recognized by the TPP. They depend on the decision that the ASPSP makes dynamically when evaluating the operation. Supported types in this response. • 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 contained under the same conditions as	Links	COND	Ex: "_links": {…}

	the "scaMethods" field			
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": […]

6.3.6.3 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 2YotnFZFEjr1zCsicMWpAA

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 where an authorization of the cancellation by the PSU is not necessary

HTTP / 1.1 204 No content

```
X-Request-ID: 0ee25bf4-6ff1-11e8-adc0-fa7ae01bbebc
Date: Sun, 26 Sep 2017 15:02:47 GMT
```

Example answer where if an implicit authorization of the cancellation by the PSU is necessary

```
HTTP / 1.1 202 Ok
X-Request-ID: 0ee25bf4-6ff1-11e8-adc0-fa7ae01bbebc
Date: Sun, 26 Sep 2017 15:02:47 GMT
{
        "transactionStatus": "ACTC",
        "_links": {
        "scaRedirect": {
                        "href": "https://api.hub.com/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/cancellation-
                authorisations/qwer-234/state"
                }
        }
}
```

Example answer where if an implicit authorization of the cancellation by the PSU is necessary

```
HTTP / 1.1 202 Ok
X-Request-ID: 0ee25bf4-6ff1-11e8-adc0-fa7ae01bbebc
Date: Sun, 26 Sep 2017 15:02:47 GMT
{
        "transactionStatus": "ACTC",
        "_links": {
                "self": {
                        "href": "/v1.1/payments/sepa-credit-transfers/123-qwe-456"
                },
                "state": {
                        "href": "/v1.1/payments/sepa-credit-transfers/123-qwe-456/state"
                },
                "startAuthorisation": {
                        "href":  "/v1.1/payments/sepa-credit-transfers/123-qwe-456/cancellation-
                authorisations"
                }
        }
}
```

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)	**Request consent on specified accounts** Create a consent, which the ASPSP must store, requesting access to the specified accounts and with the requested access. If a consent already existed, said consent will expire and the new one will take effect when authorized by the USP. 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”.
Global consent (Global consent)	**Request consent on the list of available accounts** 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". 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".

	It is a one-time consent to obtain the list of available accounts. It will not give the details of the accounts. **Request consent to access all accounts for all PSD2 AIS services** Request access for all available PSU accounts on all PSD2 AIS services. The accounts are not given by the TPP. 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". The TPP, through the HUB, can retrieve such information managed between ASPSP and PSU by making a request to retrieve consent information.
Consent offered by the bank (Bank offered consent)	**Request consent without indicating accounts** 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. 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. The ASPSP in the redirection process will show the PSU its accounts to choose which ones it wants to consent to the TPP. The TPP, through the HUB, can retrieve such information managed between ASPSP and PSU by making a request to retrieve consent information.

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.
- In either case, the final decision to return the PSU name through the API will depend on whether you are currently returning through the ASPSP's online channels.

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 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 6: SCA flow by redirection: implicit start of the authorisation processthe 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

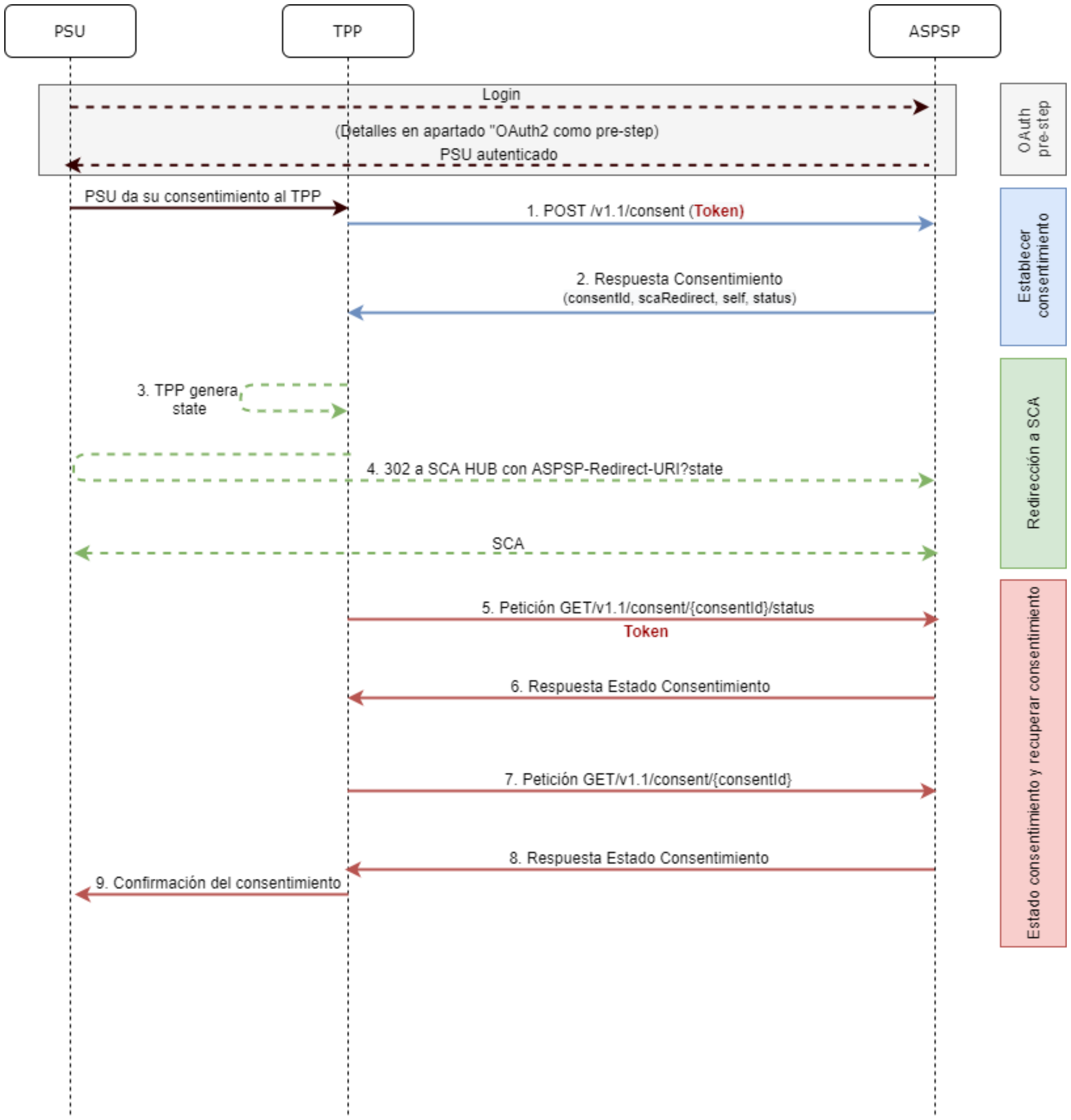

PSU
TPP
ASPSP
Login
(Detalles en apartado "OAuth2 como pre-step)
PSU autenticado
PSU da su consentimiento al TPP
1. POST /v1.1/consent (Token)
2. Respuesta Consentimiento
(consentId, scaRedirect, self, status)
3. TPP genera state
4. 302 a SCA HUB con ASPSP-Redirect-URI?state
SCA
5. Petición GET/v1.1/consent/{consentId}/status
Token
6. Respuesta Estado Consentimiento
7. Petición GET/v1.1/consent/{consentId}
8. Respuesta Estado Consentimiento
9. Confirmación del consentimiento
OAuth pre-step
Establecer consentimiento
Redirección a SCA
Estado consentimiento y recuperar consentimiento

Figure 6: 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 6: 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: implicit start of authorization process.

Similar to 6.3.1.2 SCA flow by redirection: implicit 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
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** ^[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. 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.	String	OP	Ex: PSU-Corporate-ID-Type: email

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**	Boolea n	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	^.{12,50}$ 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 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. **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
TPP-Rejection-NoFunds-Preferred	**Note:** This field will be ignored if it is reported by the TPP.	String	OP	
TPP-Notification-URI	**Note:** This field will be ignored if it is reported by the TPP.	String	OP	
TPP-Notification-Contained-Preferred	**Note:** This field will be ignored if it is reported by the TPP.	String	OP	

Body

Field	Description	Type	Mandat.	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".	AccountAccess	MA	Ex: "access": {…}
recurringIndicator	Possible values: • 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 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	Mandat.	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	Value returned if the SCA method has been set. Possible values: • EMBEDDED • DECOUPLED • REDIRECT The OAuth based SCA will be taken as REDIRECT.	String	COND	Ex: ASPSP-SCA-Approach: REDIRECT
ASPSP-Notification-Support	Not in use. Resource state notification services not supported	Boolean	NA	
ASPSP-Notification-Content	Not in use. Resource state notification services not supported	String	NA	

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"
scaMethods	This element is contained if SCA is required and if the PSU can choose between different authentication methods. If this data is contained, the link "startAuthorisationWithAuthenticationMethodSelection" will also be reported. These methods must be presented to the PSU. **Note:** Only if ASPSP supports SCA method selection	List<AuthenticationObject>	COND	Ex: "scaMethods": […]
chosenScaMethod	**NOT SUPPORTED IN THIS VERSION.**	AuthenticationObject	COND	
_links	List of hyperlinks to be recognized by the TPP. Supported types in this response: • 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	Links	OB	Ex: "_links": {…}

	generated while selecting the SCA method. This link is contained under the same conditions as the "scaMethods" field - 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	^.{1,500} $ Ex: "psuMessage": "Información para PSU"
tppMessages	Message to the TPP	List<TppMessage>	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 2YotnFZFEjr1zCsicMWpAA

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
{
        "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 2YotnFZFEjr1zCsicMWpAA
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
{
        "access": {
                "availableAccounts": "allAccounts"
        },
"recurringIndicator": false,
        "validUntil": "2018-05-17",
        "frequencyPerDay": 1
}
```

Example request consent without indicating accounts with decoupled SCA

```
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 2YotnFZFEjr1zCsicMWpAA
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: false
Date: Sun, 26 Sep 2017 15:02:37 GMT
{
    "access": {
        "balances": [],
        "transactions": []
    },
    "recurringIndicator": true,
    "validUntil": "2018-05-17",
    "frequencyPerDay": 4
}
```

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": {
            "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"
        }
    }
}
```

Example response in case of decoupled SCA

```
HTTP/1.1 201 Created
X-Request-ID: 10391c7e-ad88-49ec-a2ad-00aacb1f6541
ASPSP-SCA-Approach: DECOUPLED
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": {
        "self": {
            "href": "/v1.1/consents/123-asdf-456",
        "state": {
            "href": "/v1.1/consents/123-asdf-456/state"
        }
    },
        "psuMessage": "Please use your Bank XXX application to authorise consent".
}
```

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. Previously sent in response to a consent request message from the TPP to the HUB.	String	MA	^.{1,36}$ Ex:123-qwerty-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** ^[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** ^[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	^.{1,500} $

				Ex: "psuMessage": "Información para PSU"
tppMessages	Message to the TPP	List<Tpp Message >	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 2YotnFZFEjr1zCsicMWpAA
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"
}
```

6.4.5 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
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	Mandat.	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: • 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"
_links	Recommended link types for this response: • account Depending on the nature of the consent.	Links	OP	Ex: "_links": {…}
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 2YotnFZFEjr1zCsicMWpAA

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": [
                {
                        "iban": "ES1111111111111111111111"
                },
                {
                        "iban": "ES2222222222222222222222",
                        "currency": "USD"
                },
                {
                        "iban": "ES3333333333333333333333"
                }
                ],
                "transactions": [
                {
                        "iban": "ES1111111111111111111111"
                }
                ]
        },
        "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	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. 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 2YotnFZFEjr1zCsicMWpAA

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	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: • List of all available PSU accounts. You will not be able to obtain: • Account balances (unless supported by ASPSP) • Links to balance or transaction endpoints
availableAccountsWithBalance	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: • List of all available PSU accounts. • Account balances (unless supported by ASPSP) You will not be able to obtain: • Links to balance or transaction endpoints

account	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.
balances	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.
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	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:

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

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.	String	MA	Ex: Authorization: Bearer 2YotnFZFEjr1zCsicMWpAA
Consent-ID	Identifier of the consent obtained in the transaction to request consent.	String	MA	^.{1,36}$ Ex: Consent-ID: 7890-asdf-4321
PSU-IP-Address	IP address of the HTPP request between the PSU and the TPP. It should only be included if this request was actively initiated by the PSU.	String	COND	^[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	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

Body

Field	Description	Type	Mandat.	Format
accounts	List of available accounts.	List<AccountDetails>	MA	Ex: "accounts": []
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": […]

6.5.1.3 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 2YotnFZFEjr1zCsicMWpAA

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 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	AccountDetails	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<TppMessage>	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 2YotnFZFEjr1zCsicMWpAA

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 single-currency account 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-
                        5400a64e80f/transactions"
                        }
                }
    }
}
```

Example multi-currency account 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-f5400a64e81g",
                "iban": "ES2222222222222222222222",
                "currency": "XXX",
                "ownerName": "Heike Mustermann",
                "product": "Multicurrency Account",
                "cashAccountType": "CACC",
                "name": "Aggregation Account",
                "_links": {
                        "balances": {
                                "href": "/v1.1/accounts/3dc3d5b3-7023-4848-9853-f5400a64e81g/balances"
                        },
                        "transactions": {
                                "href": "/v1.1/accounts/3dc3d5b3-7023-4848-9853-f5400a64e81g/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	OB	Ex: www.hub.com
aspsp	Name of the ASPSP to which the request is to be made.	String	OB	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	OB	^.{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. Note: it is recommended to use it as it could become a mandatory parameter in future versions.	AccountReference	OP	Ex: "account": {…}
balances	A list of balances with respect to an account.	List<Balance>	OB	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 2YotnFZFEjr1zCsicMWpAA
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.

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	OB	Ex: www.hub.com
aspsp	Name of the ASPSP to which the request is to be made.	String	OB	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	OB	^.{1,100} $ Ex: account-id = a1q5w

Query parameters:

Field	Description	Type	Mandat.	Format
dateFrom	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. For booked transactions, the relevant date is the date of the bookingDate entry. For pending transactions, the relevant date is the entry "entryDate".	String	COND	**ISODate** Ex: dateFrom=2017-10-25

dateTo	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. For booked transactions, the relevant date is the date of the bookingDate entry. For pending transactions, the relevant date is the entry "entryDate".	String	OP	**ISODate** Ex: dateTo=2017-11-05
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: • booked (OB) • pending (OP) • both (OP) **Note:** pending and both only if they are supported by ASPSP. Additionally, the state is supported: • Information (OB) To return the list of standing orders.	String	OB	Ex: bookingStatus = booked
deltaList	Indicates that the AISP is in favour of obtaining all transactions after the last report access for this PSU and account.	Boolean	OP	Ex: deltaList = false

	This indicator may be rejected by the ASPSP if this function is not supported.			
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

Note: in case bookingStatus is equal to "information", the query param dateFrom, dateTo, withBalance, deltaList and entryReferenceFrom 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	Mandat.	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.	AccountReference	OP	Ex: "account": {…}

transactions	Return of data in JSON format, when the returned data are small in length.	AccountReport	OP	Ex: "transactions": {…}
balances	A list of balances with respect to an account.	List<Balance>	OP	Ex: "balances": {…}
_links	List of hyperlinks to be recognized by the TPP. Supported types in this response: "**download**": Link to download the data of the query performed, when the returned data are oversized. Only for camt-data.	Links	OP	Ex: "_links": {…}
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": […]

6.5.4.3 Examples

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

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 2YotnFZFEjr1zCsicMWpAA

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 2YotnFZFEjr1zCsicMWpAA

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 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",
                        Ex: "debtorName": "Paul Simpson"
                        "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": {
        "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 2YotnFZFEjr1zCsicMWpAA

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 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-STDO",
                        "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_EXCEDED "
        }
    ]
}
```

6.6 FCS: Establish consent for funds confirmation service

6.6.1 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.

6.6.1.1 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	OB	Ex: www.hub.com
aspsp	Name of the ASPSP to which the request is to be made.	String	OB	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	OB	**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
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.	String	OP	Ex: PSU-Corporate-ID-Type: email
Authorization	Bearer Token. Obtained in a previous authentication on OAuth2.	String	OB	Ex: Authorization: Bearer 2YotnFZFEjr1zCsicMWpAA
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.	Boolean	OP	Ex: TPP-Redirect-Preferred: true

	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**			
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. The domain of this URI is required to be the same as the content in the TPP web certificate.	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. The domain of this URI is required to be the same as the content in the TPP web certificate.	String	OP	^.{12,50}$ 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.	Boolean	OP	Ex: TPP-Explicit-Authorisation-Preferred: false

	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.			
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
TPP-Rejection-NoFunds-Preferred	**Note:** This field will be ignored if it is reported by the TPP.	String	OP	
TPP-Notification-URI	**Note:** This field will be ignored if it is reported by the TPP.	String	OP	
TPP-Notification-Contained-Preferred	**Note:** This field will be ignored if it is reported by the TPP.	String	OP	

Body

Field	**Description**	**Type**	**Mandat.**	**Format**
account	Account on which the fund consultation is to be carried out.	AccountReference	OB	Ex: "access": {…}
cardNumber	Card number of the card issued by the PIISP. Must be sent if available.	String	OP	^.{1,35}$
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}$

6.6.1.2 Response

HTTP Code

201 if the resource has been created

Response code

HTPP 201 response code if resource is created successfully.

Header

Field	Description	Type	Mandat.	Format
Location	Contains the hyperlink to the generated resource	String	OB	**Max512Text** Ex: Location: /v2.1/consents/confirmation-of-funds/{consentId}
X-Request-ID	Unique identifier of the operation assigned by the TPP.	String	OB	**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: • EMBEDDED • DECOUPLED • REDIRECT	String	COND	Ex: ASPSP-SCA-Approach: REDIRECT

	The OAuth based SCA will be taken as REDIRECT.			
ASPSP-Notification-Support	Not in use. Resource state notification services not supported	Boolean	NA	
ASPSP-Notification-Content	Not in use. Resource state notification services not supported	String	NA	

Body

Field	Description	Type	Mandat.	Format
consentStatus	Consent State Defined values in 9.5 Consent states	String	OB	Ex: "consentStatus": "received"
consentId	Identifier of the resource that refers to the consent. It must be contained if consent was generated.	String	OB	^.{1,36}$ Ex: "consentId": "123-QWE-456"
scaMethods	This element is contained if SCA is required and if the PSU can choose between different authentication methods. If this data is contained, the link "startAuthorisationWithAuthenticationMethodSelection" will also be reported. These methods must be presented to the PSU. **Note:** Only if ASPSP supports SCA method selection	List<AuthenticationObject>	COND	Ex: "scaMethods": […]
_links	List of hyperlinks to be recognized by the TPP. Supported types in this response: • scaRedirect: in case of SCA by redirection. Link where the PSU	Links	OB	Ex: "_links": {…}

	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 contained under the same conditions as the "scaMethods" field - 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	^.{1,500} $ Ex: "psuMessage": "Información para PSU"
tppMessages	Message to the TPP	List<TppMessage>	OP	Ex: "tppMessages": […]

6.6.1.3 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 2YotnFZFEjr1zCsicMWpAA
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
{
        "account": {
                "iban": "ES1111111111111111111111"
        },
        "cardNumber": "123456781234",
        "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"
                }
        }
}
```

Example response in case of decoupled SCA

```
HTTP/1.1 201 Created
X-Request-ID: 10391c7e-ad88-49ec-a2ad-00aacb1f6541
ASPSP-SCA-Approach: DECOUPLED
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": {
                "self": {
                        "href": "/v2.1/consents/confirmation-of-funds/123-asdf-456",
                "state": {
                        "href": "/v2.1/consents/confirmation-of-funds/123-asdf-456",
                }
        },
        "psuMessage": "Please use your Bank XXX application to authorise consent".
}
```

6.6.2 Obtain consent state

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

6.6.2.1 Request

Endpoint

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

Path

Field	Description	Type	Mandat.	Format
provider	URL of the HUB where the service is released.	String	OB	Ex: www.hub.com
aspsp	Name of the ASPSP to which the request is to be made.	String	OB	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	OB	^.{1,36}$ Ex:123-qwerty-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	OB	**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	OB	Ex: Authorization: Bearer 2YotnFZFEjr1zCsicMWpAA

Body

No additional data travels.

6.6.2.2 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	OB	**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	Mandat.	Format
consentStatus	Consent authentication state. Defined values in 9.5 Consent states	String	OB	Ex: "consentStatus":"valid"
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": […]

6.6.2.3 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 2YotnFZFEjr1zCsicMWpAA

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"
}
```

6.6.3 Retrieve consent information

6.6.3.1 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	Mandat.	Format
provider	URL of the HUB where the service is released.	String	OB	Ex: www.hub.com
aspsp	Name of the ASPSP to which the request is to be made.	String	OB	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	OB	^.{1,36}$ Ex: 7890-asdf-4321

Query parameters:

No additional fields are specified.

Header

The same as those defined in the section 6.6.2.1

Body

No additional data travels.

6.6.3.2 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 6.6.2.2

Body

Field	Description	Type	Mandat.	Format
account	Account on which the fund consultation is to be carried out.	AccountReference	OB	Ex: "access": {…}
cardNumber	Card number of the card issued by the PIISP. Must be sent if available.	String	OP	^.{1,35}$
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	OB	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<TppMessage>	OP	Ex: "tppMessages": […]

6.6.3.3 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 2YotnFZFEjr1zCsicMWpAA

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": {
                "iban": "ES1111111111111111111111"
        },
        "cardNumber": "123456781234",
        "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"
}
```

6.6.4 Revoke consent

6.6.4.1 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
provider	URL of the HUB where the service is released.	String	OB	Ex: www.hub.com
aspsp	Name of the ASPSP to which the request is to be made.	String	OB	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	OB	^.{1,36}$ Ex: 7890-asdf-4321

Query parameters:

No additional fields are specified.

Header

The same as those defined in the section 6.6.2.1

Body

No additional data travels.

6.6.4.2 Response

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

Response code

HTPP 204 response code for successful cancellation.

Header

The same as those defined in the section 6.6.2.2

Body

No additional fields are specified.

6.6.4.3 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 2YotnFZFEjr1zCsicMWpAA

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

6.7 FCS: Fund Confirmation Service

6.7.1 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.

Rules that apply to the confirmation of funds in multi-currency accounts

- If the "cardNumber" is not given, but the identifier of the PSU account is contained → -->Check the default account registered by the client
- If no "cardNumber" is given, but the PSU account identifier together with the currency is contained → -->Check availability of funds on the sub-account specified by the id+moneda
- If the "cardNumber" and the PSU account identifier is given → Check the availability of funds on the sub-account represented by the "cardNumber".
- If the "cardNumber" is not registered for any of the subaccounts, or if the "cardNumber" is registered for a different subaccount, the "cardNumber" could be ignored.

6.7.1.1 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	OB	Ex: www.hub.com
aspsp	Name of the ASPSP to which the request is to be made.	String	OB	Ex: aspsp-name

Header

Field	Description	Type	Mandat.	Format
X-Request-ID	Unique identifier of the operation assigned by the TPP.	String	OB	**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	OB	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	OB	^.{1,36}$ Ex: Consent-ID: 7890-asdf-4321

Body

Field	Description	Type	Mandat.	Format

cardNumber	Card numbering issued by PIISP. Must be sent if available.	String	OP	Ex: "cardNumber": "1111-1111-1111-1111"
account	PSU account number.	AccountReference	OB	Ex: "account": {"iban":"ES1111111111111111111"}
payee	Commerce where the card is accepted as information for the PSU.	String	OP	^.{1,70}$ Ex: "payee":"Nombre comercio"
instructedAmount	Contains the amount and currency to consult.	Amount	OB	Ex: "instructedAmount": {…}

6.7.1.2 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	OB	**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	Mandat.	Format

fundsAvailable	It takes the value "true" if there are sufficient funds available at the time of the request; "false" otherwise.	Boolean	OB	Ex: "fundsAvailable": true
tppMessages	Message to the TPP	List<TppMessage>	OP	Ex: "tppMessages": […]

6.7.1.3 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 2YotnFZFEjr1zCsicMWpAA
Consent-ID: 7890-asdf-4321
Date: Sun, 17 Oct 2017 13:15:17 GMT
{
        "cardNumber": "87432569872156",
        "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
}
```

6.8 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.

6.9 Processes common to services

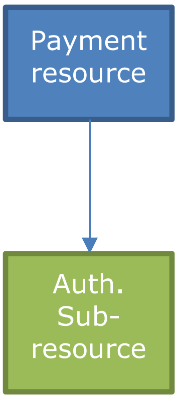
6.9.1 Start the authorization process (explicit)

Use

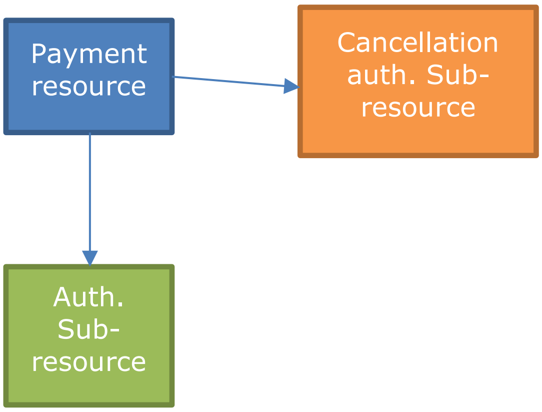
The initiate authorization process is a process required to create a new authorization sub-resource (if not created implicitly). Applies in the following scenarios:

- 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 response to a request to cancel payment that an explicit start of the authorization 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.

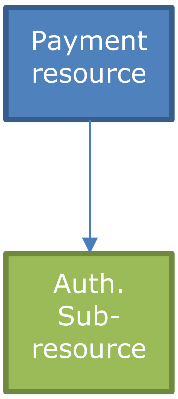
Abstract data model



Data model for a planned payment with a single SCA



Data model for a planned payment which has been canceled, where the client's authorization for cancellation has been required



Data model for a planned payment which has been cancelled, where no client authorisation was required for the cancellation.

The payment state remains as CANC and the resource is no longer routable.

6.9.1.1 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 Payment Cancellation

POST {provider}/{aspsp}/v1.1/{payment-service}/{payment-product}/{paymentId}/cancellation-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	OB	Ex: hub.example.es
aspsp	Name of the ASPSP to which the request is to be made.	String	OB	Ex: aspsp-name
payment-service	Possible values are: • payments • periodic-payments	String	COND	Ex: {provider}/v1.1/payments
payment-product	Paid product to use. List of supported products: • sepa-credit-transfers • instant-sepa-credit-transfers • target-2-payments • cross-border-credit-transfers	String	COND	Ex: {provider}/v1.1/payments/sepa-credit-transfers/
paymentI, consentId	Resource identifier that refers to the initiation of payment.	String	OB	^.{1,36}$ Ex: 123-qwe-456

Query parameters:

No additional parameters are specified for this request.

Header

Field	Description	Type	Mandat.	Format

Content-Type	Value: application / json	String	OB	Content-Type: application/json
X-Request-ID	Unique transaction identifier assigned by the TPP and forwarded via the HUB to the ASPSP	String	OB	**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	OB	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	If "true", the TPP has communicated to the HUB that it prefers SCA by redirection.	Boolean	OP	Ex: TPP-Redirect-Preferred: true

	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**			
TPP-Redirect-URI	HUB URI where the flow of the transaction should be redirected after finishing the SCA by redirect. 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://hub.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://hub.example.es/cb/nok"

Body

No additional fields are specified.

6.9.1.2 Response

HTTP Code

201 if the resource has been created

Header

Field	Description	Type	Mandat.	Format
Location	Contains the link to the generated resource.	String	OB	Ex: Location: /v1.1/payments/{payment-product}/{paymentId}/authorisations/123qwert/456
X-Request-ID	Unique identifier of the operation assigned by the TPP and sent through the HUB to the ASPSP.	String	OB	**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: • EMBEDDED • DECOUPLED • REDIRECT The SCA based on OAuth2 will be taken as REDIRECT.	String	COND	Ex: ASPSP-SCA-Approach: REDIRECT

Body

Field	Description	Type	Mandat.	Format
scaStatus	SCA state	String	OB	Ex: "scaStatus": "received"
authorizationId	Resource identifier that refers to the authorization sub-resource created.	String	OB	^.{1,36}$ Ex: "authorisationId": "1b3ab8e8-0fd5-43d2-946e-d75958b172e7"

scaMethods	This element is contained if SCA is required and if the PSU can choose between different authentication methods. If this data is contained, the "selectAuthenticationMethod" link will also be reported. These methods must be presented to the PSU. **Note:** Only if ASPSP supports SCA method selection	List<Authentication Object>	COND	Ex: "scaMethods": […]
_links	List of hyperlinks to be recognized by the TPP. Supported types in this response: • scaRedirect: in case of SCA by redirection. Link where the PSU browser must be redirected by the TPP. • selectAuthenticationMethod: link to the authorization or cancellation authorization sub-resource where the selected SCA method will be informed. • scaStatus: link to consult the SCA state corresponding to the authorisation sub-resource.	Links	OB	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<TppMessage>	OP	Ex: "tppMessages": […]

6.9.1.3 Examples

Example request about a Payment Cancellation

POST https://hub.example.es/aspsp-name/v1.1/payments/sepa-credit-transfers/qwert1234tzui7890/cancellation-authorisations

Content-Encoding: gzip

Content-Type: application/json

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

Authorization: Bearer 2YotnFZFEjr1zCsicMWpAA

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

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

Example response in case of SCA by redirection

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/cancellation-authorisations/123auth456

Content-Type: application/json

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

6.9.2 Get authorization sub-resources

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

6.9.2.1 Request

Endpoint in case of Start of Payment
GET {provider}/{aspsp}/v1.1/{payment-service}/{payment-product}/{paymentId}/authorisations

Endpoint in case of Payment Cancellation

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

Endpoint in case of Account Information Consent
GET {provider}/{aspsp}/v1.1/consents/{consentId}/authorisations

Path

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

payment-service	Possible values are: • payments • periodic-payments	String	COND	Ex: {provider}/v1.1/payments
payment-product	Paid product to use. List of supported products: • 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	OB	^.{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.

6.9.2.2 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**	**Mandat.**	**Format**

authorisationIds	Array of authorizationIds. **Note**: required field if it is not a cancellation	Array<String>	COND	^.{1,36}$ Ex: " authorisationIds ": […]
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": […]

6.9.2.3 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 2YotnFZFEjr1zCsicMWpAA

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"]
}
```

6.9.3 Get SCA state

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

6.9.3.1 Request

Endpoint in case of Start of Payment

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

Endpoint in case of Payment Cancellation

GET {provider}/{aspsp}/v1.1/{payment-service}/{payment-product}/{paymentId}/cancellation-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	OB	Ex: hub.example.es
aspsp	Name of the ASPSP to which the request is to be made.	String	OB	Ex: aspsp-name
payment-service	Possible values are: • payments • periodic-payments	String	COND	Ex: {provider}/v1.1/payments
payment-product	Paid product to use. List of supported products: • 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	OB	^.{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.

6.9.3.2 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	Mandat.	Format
scaStatus	SCA state	String	OB	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: ‘trustedBeneficiary Flag’: 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": […]

6.9.3.3 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 2YotnFZFEjr1zCsicMWpAA

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

```
{
	"scaStatus": " finalised"
}
```

7. DESCRIPTION SERVICES OF ADDED VALUE

7.1 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

7.1.1 Payment Initiation Flows

7.1.1.1 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.

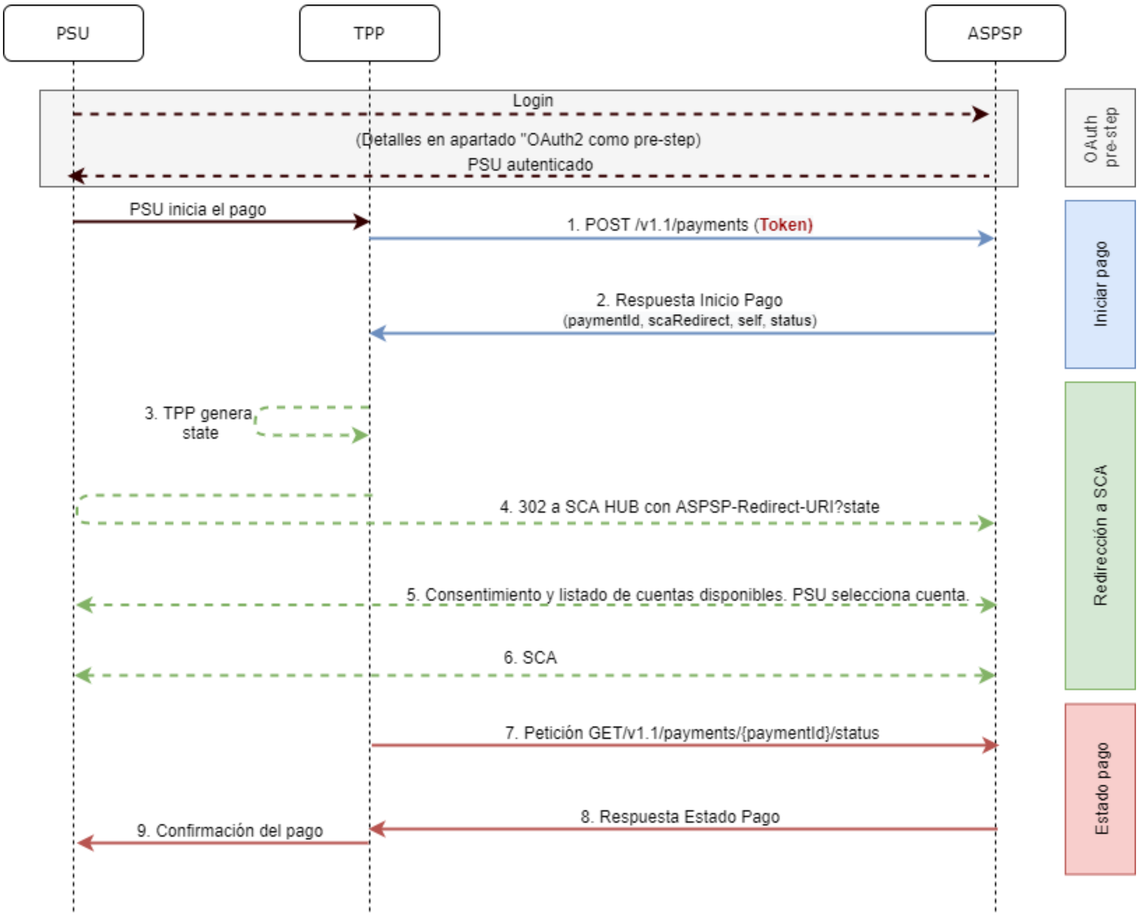

PSU
TPP
ASPSP
Login
(Detalles en apartado "OAuth2 como pre-step)
PSU autenticado
OAuth pre-step
PSU inicia el pago
1. POST /v1.1/payments (Token)
2. Respuesta Inicio Pago
(paymentId, scaRedirect, self, status)
Iniciar pago
3. TPP genera state
4. 302 a SCA HUB con ASPSP-Redirect-URI?state
5. Consentimiento y listado de cuentas disponibles. PSU selecciona cuenta.
6. SCA
Redirección a SCA
7. Petición GET/v1.1/payments/{paymentId}/status
8. Respuesta Estado Pago
9. Confirmación del pago
Estado pago

Figure 7: 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 4: 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.

7.1.1.1 SCA flow by redirection: implicit start of authorization process

Similar to6.3.1.2 SCA flow by redirection: implicit start of authorization.

7.1.2 Payment initiation completion

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

7.1.2.1 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	OB	Ex: www.hub.com
aspsp	Name of the ASPSP to which the request is to be made.	String	OB	Ex: aspsp-name
payment-product	Paid product to use. List of supported products: • sepa-credit-transfers • instant-sepa-credit-transfers • target-2-payments • cross-border-credit-transfers	String	OB	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 8.16 SinglePayment following the conditions of the following table.

The fields marked as mandatory (OB) and optional (OP) are supported by the ASPSP with this type of condition.

The fields marked as COND depend on each ASPSP.

Field	SCT	SCT INST	Target 2	Cross Border CT
endToEndIdentification	OP	OP	OP	COND
instructionIdentification	COND	COND	COND	COND

debtorName	COND	COND	COND	COND
debtorAccount	NA	NA	NA	NA
debtorId	COND	COND	COND	COND
ultimateDebtor	COND	COND	COND	COND
instructedAmount	MA	MA	MA	MA
currencyOfTransfer	COND	COND	COND	COND
exchangeRateInformation	COND	COND	COND	COND
creditorAccount	MA	MA	MA	MA
creditorAgent	OP	OP	OP	OB/OP
creditorAgentName	COND	COND	COND	COND
CreditorName	MA	MA	MA	MA
creditorId	COND	COND	COND	COND
creditorAddress	OP	OP	OP	OP
creditorNameAndAddress	COND	COND	COND	COND
ultimateCreditor	COND	COND	COND	COND
purposeCode	COND	COND	COND	COND
chargeBearer	COND	COND	COND	COND
serviceLevel	COND	COND	COND	COND
remittanceInformationUnstructured	OP	OP	OP	OP
remittanceInformationUnstructuredArray	COND	COND	COND	COND
remittanceInformationStructured	COND	COND	COND	COND
remittanceInformationStructuredArray	COND	COND	COND	COND
requestedExecutionDate	n.a.	n.a.	n.a.	n.a.
requestedExecutionTime	n.a.	n.a.	n.a.	n.a.

7.1.2.2 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

7.1.2.3 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 2YotnFZFEjr1zCsicMWpAA
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": "ES2222222222222222222222"
        },
        "creditorName": "Nombre123",
        "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"
                }
        }
}
```

7.2 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

7.2.1 Periodic 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.

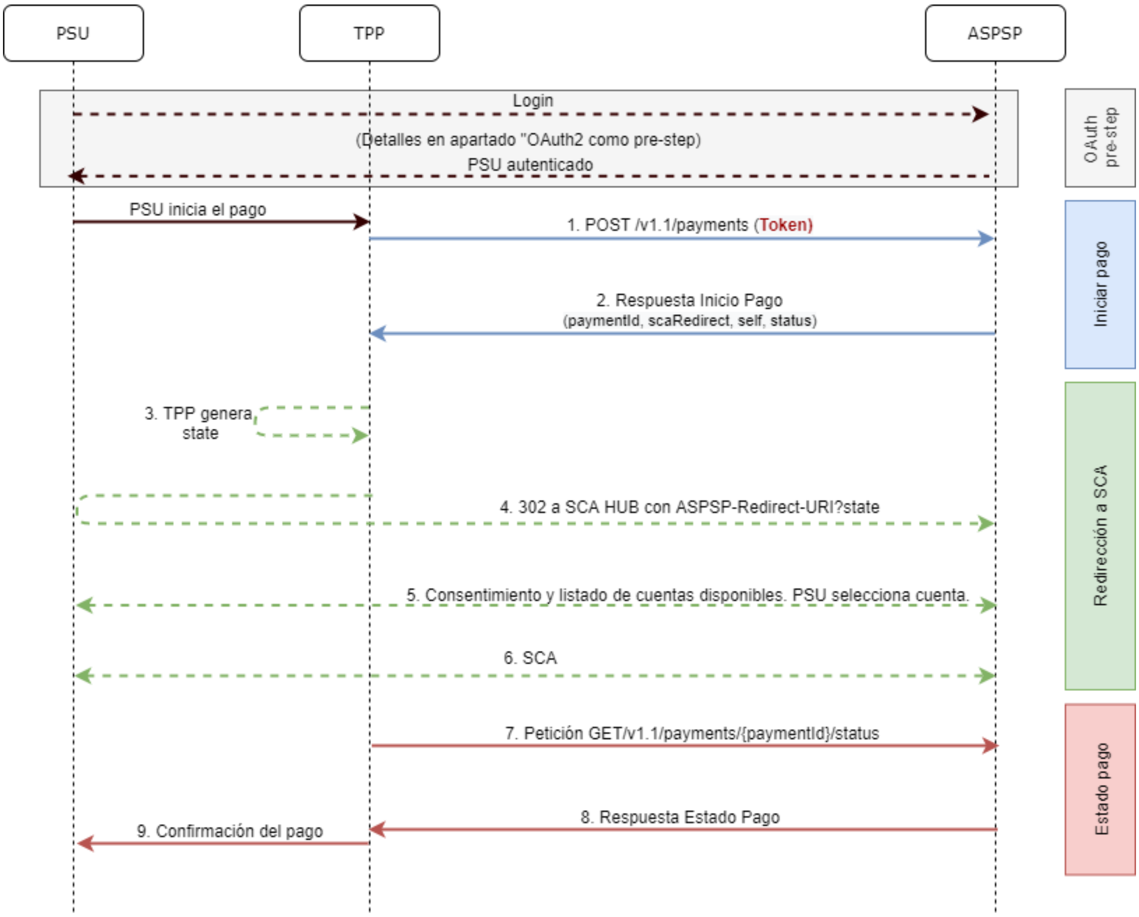

PSU
TPP
ASPSP
Login
(Detalles en apartado "OAuth2 como pre-step)
PSU autenticado
OAuth pre-step
PSU inicia el pago
1. POST /v1.1/payments (Token)
2. Respuesta Inicio Pago
(paymentId, scaRedirect, self, status)
Iniciar pago
3. TPP genera state
4. 302 a SCA HUB con ASPSP-Redirect-URI?state
5. Consentimiento y listado de cuentas disponibles. PSU selecciona cuenta.
6. SCA
Redirección a SCA
7. Petición GET/v1.1/payments/{paymentId}/status
8. Respuesta Estado Pago
9. Confirmación del pago
Estado pago

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.

7.2.1.2 SCA flow by redirection: implicit start of authorization process

Similar to6.3.1.2 SCA flow by redirection: implicit start of authorization.

7.2.2 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

- **Daily payments**: the "dayOfExecution" field is not necessary. The first payment is the "startDate" and, from there, the payment is made every day
- **Weekly payments**: if "dayOfExecution" is required, the possible values are from 01 = Monday to 07 = Sunday. If "dayOfExecution" is not required, "startDate" is taken as the day of the week the payment is made. (If "startDate" is Thursday, the payment would be made every Thursday)
- **Bi-weekly payments**: same rule applies as weekly payments.
- **Monthly payments or higher:** possible values range from 01 to 31. Using 31 as the last day of the month

7.2.2.1 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	OB	Ex: aspsp.example.es
payment-product	Paid product to use. List of supported products: • sepa-credit-transfers • instant-sepa-credit-transfers • target-2-payments • cross-border-credit-transfers	String	OB	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 Error! Reference source not found**¡Error! No se encuentra el origen de la referencia. ¡Error! No se encuentra el origen de la referencia.**, 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	OB	**ISODate** xEx: "startDate":"2018-12-20"
executionRule	Supported values: • 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. The ASPSP may reject the request due to the communicated value if the Online Banking rules do not support this execution rule.	String	OP	Ex: "executionRule":"following"
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: • Daily • Weekly • EveryTwoWeeks • Monthly • EveryTwoMonths • Quarterly	String	OB	**EventFrequency7Code de ISO 20022** Ex: "frequency": "Monthly"

	- Semi Annual - Annual			
dayOfExecution	"31" is last. Follows the regular expression\d{1,2} The date refers to the ASPSP time zone. Only if supported in ASPSP Online Banking.	String	COND	\d{1,2} Ex: "dayOfExecution": "01"

The fields marked as mandatory (MA) and optional (OP) are supported by the ASPSP with this type of condition.

The fields marked as COND depend on each ASPSP.

Field	**SCT**	**SCT INST**	**Target 2**	**Cross Border CT**
EndToEndIdentification*	NA	NA	NA	NA
instructionIdentification	COND	COND	COND	COND
debtorName	COND	COND	COND	COND
debtorAccount	NA	NA	NA	NA
debtorId	COND	COND	COND	COND
ultimateDebtor	COND	COND	COND	COND
instructedAmount	MA	MA	MA	MA
currencyOfTransfer	COND	COND	COND	COND
exchangeRateInformation	COND	COND	COND	COND
creditorAccount	MA	MA	MA	MA
creditorAgent	OP	OP	OP	OB/OP
creditorAgentName	COND	COND	COND	COND
CreditorName	MA	MA	MA	MA
creditorId	COND	COND	COND	COND
creditorAddress	OP	OP	OP	OP
creditorNameAndAddress	COND	COND	COND	COND
ultimateCreditor	COND	COND	COND	COND

purposeCode	COND	COND	COND	COND
chargeBearer	COND	COND	COND	COND
serviceLevel	COND	COND	COND	COND
remittanceInformationUnstructured	OP	OP	OP	OP
remittanceInformationUnstructuredArray	COND	COND	COND	COND
remittanceInformationStructured	COND	COND	COND	COND
remittanceInformationStructuredArray	COND	COND	COND	COND
requestedExecutionDate	n.a.	n.a.	n.a.	n.a.
requestedExecutionTime	n.a.	n.a.	n.a.	n.a.

*** NOTE: If the TPP wants to inform it, it will travel in the remittanceInformationUnstructured field, providing it with a good practice guide for its use.**

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

Body

Field	Description	Type	Man dat.	Format
transactionStatus	Transaction state. Values defined in annexes in **¡Error! No se encuentra el origen de la referencia. ¡Error! No se encuentra el origen de la referencia.**	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. If it is equal to "false" or not used, the transaction will not involve any additional fees for the PSU.	Boolean	OP	Ex: "transactionFeeIndicator": true
scaMethods	This element is contained if SCA is required and if the PSU can choose between different authentication methods. If this data is contained, the link "startAuthorisationWithAuthenticationMethodSelection" will also be reported. These methods must be presented to the PSU.	List<AuthenticationObject>	COND	Ex: "scaMethods": […]
chosenScaMethod	**NOT SUPPORTED IN THIS VERSION.**	Authentication Object	COND	
_links	List of hyperlinks to be recognized by the TPP. Supported types in this response: • 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.	Links	OB	Ex: "_links": {…}

	• self: link to the payment initiation resource created by this request. • 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	^.{1,500} $ Ex: "psuMessage": "Información para PSU"
tppMessages	Message for the TPP sent through the HUB.	List<TppMessage>	OP	Ex: "tppMessages" : […]

7.2.2.3 Examples

Example of request

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

Content-Encoding: gzip

Content-Type: application/json

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 2YotnFZFEjr1zCsicMWpAA

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": "ES2222222222222222222222"
        },
        "creditorName": "Nombre123",
        "remittanceInformationUnstructured": "Información adicional",
        "startDate": "2018-03-01",
        "executionRule": "preceding",
        "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
ASPSP-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	Mandat.	Format
accounts	Indicates the accounts on which to request detailed information.	List<AccountReference>	OP	Ex: "accounts": […]

	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.			
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. 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.	List<AccountReference>	OP	Ex: "transactions": {…}
additionalInformation	**Note**: the information contained in this object will be ignored by the ASPSP.	AdditionalInformationAccess	OP	Ex: "additionalInformation": {…}
availableAccounts	Only the value "allAcounts" is allowed	String	OP	Ex: "availableAccounts": "allAcounts"
availableAccountsWithBalance	Only the value "allAcounts" is allowed	String	OP	Ex: "availableAccountsWithBalance": "allAcounts"

allPsd2	Only the value "allAcounts" is allowed	String	OP	Ex: "allPsd2": "allAcounts"

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":"ES1111111111111111111"
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"
state	State of the account. The value is one of the following: • 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":"XSXHXSMMXXX"
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: • PRIV: private personal account • ORGA: professional account	String	OP	^.{1,4}$ Ex: "usage": "PRIV"

details	Specifications that must be provided by the ASPSP. • Features of the account • Characteristics of the card	String	OP	^.{1,500} $
balances	Account balances.	List<Balances>	COND	"balances": […]
_links	Links to the account to retrieve account balance and/or transaction information. Links supported only when the corresponding consent to the account has been given.	Links	OP	Ex: "links": {…}

8.3 AccountReference

Field	**Description**	**Type**	**Mandat.**	**Format**
iban	Account IBAN	String	COND	Ex: "iban":"ES1111111111111111111"
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	Mandat.	Format
booked	Last known transactions (annotations) of the account. It must be included if the bookingStatus parameter is set to “booked” or “both”.	List<Transactions>	COND	Ex: "booked":[{..}]
pending	Pending account transactions. Not contained if the bookinStatus parameter is set to “booked”.	List<Transactions>	OP	Ex: "pending":[{..}]
information	List of standing orders Included if the bookingStatus parameter is set to "information".	List<Transactions>	OP	Ex: "information": [{…}]
_links	The following links are accepted in this object: • account (OB) • first (OP) • next (OP) • previous (OP) • last (OP)	Links	OB	Ex: "_links":[{..}]

8.5 AdditionalInformationAccess

Field	Description	Type	Mandat.	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	String	MA	**ISO 4217**

	The decimal separator is the period.			Ex: "amount":"500.00"

8.8 AuthenticationObject

Field	Description	Type	Mandat.	Format
authenticationType	Authentication method type. Possible values: • SMS_OTP • CHIP_OTP • PHOTO_OTP • PUSH_OTP See annex 9.6 9.6 Authentication types 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. If the TPP has it available, it must be presented to the PSU.	String	MA	Ex: "name":"SMS OTP al teléfono 666777888"
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	OB	Ex: "bic":" XXXXXXXXXXX"
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	OB	Ex: "balanceAmount": {…}
balanceType	Balance type. Values supported in annex 9.7 Types of balances	String	OB	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	Mandat.	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	Mandat.	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": {…}
transaction Details	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 PaymentExchangeRate

Field	**Description**	**Type**	**Mandat.**	**Format**

unitCurrency	Currency in which the exchange rate is expressed in foreign currency. In the following example 1EUR = xxxCUR, the currency unit is the EUR.	String	OP	**ISO 4217** Ex: "unitCurrency": "EUR"
exchangeRate	Factor used to convert an amount from one currency to another. Reflects the price at which a currency was acquired with another currency.	String	OP	Ex: "exchangeRate": "1.3"
contractIdentification	Unique identification to identify the currency exchange contract	String	OP	Ex: "contractIdentification": "1234-qeru-23"
rateType	Specifies the type used to complete the currency exchange. Allowed values: • SPOT • SALE • AGRD	String	OP	Ex: "rateType": "SPOT"

8.15 ReportExchangeRate

Field	Description	Type	Mandat.	Format
sourceCurrency	Currency from which an amount will be converted into a currency conversion	String	MA	**ISO 4217** Ex: "sourceCurrency": "EUR"
exchangeRate	Factor used to convert an amount from one currency to another. Reflects the price at which a currency was acquired with another currency.	String	MA	Ex: "exchangeRate": "1.3"

unitCurrency	Currency in which the exchange rate is expressed in foreign currency. In the following example 1EUR = xxxCUR, the currency unit is the EUR.	String	MA	**ISO 4217** Ex: "unitCurrency": "EUR"
targetCurrency	Currency in which an amount is to be converted in a currency conversion.	String	MA	**ISO 4217** Ex: "targetCurrency": "USD"
quotationDate	Date an exchange rate is quoted.	String	MA	**ISODate** Ex: "quotationDate": "2019-01-24"
contractIdentification	Unique identification to identify the currency exchange contract	String	OP	Ex: "contractIdentification": "1234-qeru-23"

8.16 SinglePayment

Field	Description	Type	Format
endToEndIdentification	Unique identifier of the operation assigned by the initiating party (TPP)	String	^.{1,35}$ Ex: "endToEndIdentification": "12345678901234567890123456789012345"
instructionIdentification	NA	NA	NA
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":"ES1111111111111111111111"}
debtorId	**NA**	String	^.{1,35}$
ultimateDebtor	**NA**	String	^.{1,70}$

instructedAmount	Information on the transfer made.	Amount	Ex: "instructedAmount": {…}
currencyOfTransfer	**NA**	String	**CurrencyCode**
exchangeRateInformation		PaymentExchangeRate	
creditorAccount	Beneficiary Account	AccountReference	Ex: ‘creditorAccount’: {‘iban’: ‘ES11111111111111111’}
creditorAgent	BIC of the beneficiary&#39;s account.	String	Ex: ‘creditorAgent’: ‘XSXHXSMMXXX’
creditorAgentName	**NA**	String	^.{1,140}$
CreditorName	Beneficiary name	String	^.{1,70}$ Ex: ‘creditorName’: ‘Name’
creditorId	**NA**	String	^.{1,35}$
creditorAddress	Beneficiary Address	Address	Ex: ‘creditorAddress’: {…}
creditorNameAndAddress		String	^.{1,140}$
ultimateCreditor	**NA**	String	^.{1,70}$
purposeCode	**NA**	String	**ExternalPurpose1Code ISO 20022**
chargeBearer	Only for payment-product: • target-2-payments • cross-border-credit-transfers Allowed values: • DEBT • CRED • SHAR • SLEV	String	**ChargeBearerType1Code from ISO 20022** Ex: ‘chargeBearer’: ‘SLEV’

serviceLevel		String	
remittanceInformationUnstructured	Additional information: See Annex9.10Guide of good practice Campo remittanceInformationUnstructured for recommendations for use.	String	^.{1,140}$ Ex: ‘remittanceInformationUnstructured’: ‘Additional information’
remittanceInformationUnstructuredArray	**NA**	List<String>	^. {1,140} $ per String
remittanceInformationStructured	**NA**	Remittance	
remittanceInformationStructuredArray	**NA**	List<Remittance>	
requestedExecutionDate	Execution date	String	**ISODate** Ex: "requestedExecutionDate": "2018-05-17"
requestedExecutionTime	Date/time excecuted	String	**ISODateTime**

8.17 StandingOrderDetails

Field	Description	Type	Mandat.	Format
startDate	The first applicable day of execution from this date is the first payment	String	OB	**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: • 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. The ASPSP may reject the request due to the communicated value if the Online Banking rules do not support this execution rule.	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. If it is equal to true, then the execution rule is overridden if the execution day falls in a different month. Note: this attribute is rarely used.	Boolean	OP	Ex: 'withinAMonthFlag': true
frequency	The frequency of the recurring payment resulting from this standing order. Allowed values: • Daily • Weekly • EveryTwoWeeks • Monthly • EveryTwoMonths • Quarterly • Semi Annual	String	OB	**EventFrequency7Code de ISO 20022** Ex: "frequency": "Monthly"

	Annual			
monthsOfExecution	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. The attribute is contained if and only if the frequency equals "MonthlyVariable".	List<String>	COND	Ex: "monthsOfExecution": ["1", "4", "10"]
multiplier	Frequency multiplier. For example, frequency = weekly and multiplier = 3 means every 3 weeks. **Note**: this attribute is rarely used	Integer	OP	Ex: "multiplier": 3
dayOfExecution	"31" is last. Following the regular expression \d{1,2} The date refers to the ASPSP time zone. Only if supported in ASPSP Online Banking.	String	COND	\d{1,2} Ex: "dayOfExecution": "01"
limitAmount	Limit amount for **Restrictions**: transactionAmount must be zero and bankTransactionCode must specify PMNT-MCOP-OTHR for	Amount	COND	Ex: "limitAmount": {…}

8.18 StructuredAdditionalInformation

Field	Description	Type	Mandat.	Format
standingOrderDetails	Standing order details	StandingOrderDetails	OB	Ex: "standingOrderDetails": {…}

8.19 TppMessage

Field	Description	Type	Mandat.	Format
category	Category of the type of message received. Possible values: ERROR or WARNING	String	OB	Ex: "category": "ERROR"
code	Response code. All return codes by service 9.3 Return Codes are listed in the annex 9.3.	String	OB	Ex: "code": "CONSENT_INVALID"
path	Path to the field referencing the error.	String	COND	Ex: "path":"…"
text	Additional explanatory text.	String	OP	Ex: "text": "Text example"

8.20 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": "..."
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 '
transaction Amount	Transaction amount	Amount	OB	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&#39;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"
remittanceInformationUnstructuredArray	Note: in version 2 of the standard the two remittanceUnstructured could be merged into one	List<String>	OP	^. {1,140} $ per String Ex: "remittanceInformationUnstructuredArray":["info1", "info2"]
remittanceInformationStructured	Field to include a reference to the sending.	String	OP	^.{1,140}$ Ex: "remittanceIinformationStructured": "Ref. 12344567 "
remittanceInformationStructuredArray	Note: in version 2 of the standard the two remittanceUnstructured could be merged into one	List<String>	OP	^. {1,140} $ per String Ex: "remittanceInformationStructuredArray":["info1", "info2"]
additionalInformation	Used by the TPP to carry additional information from the PSU	String	OP	^.{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.	Structured AdditionalInformation	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: • "PMNT-ICDT-STDO" for credit transfers, • "PMNT-IRCT-STDO" for instant credit transfers • "PMNT-ICDT-XBST" for cross-border credit transfers • "PMNT-IRCT-XBST" for cross-border real time credit transfers "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. This field is formed by concatenating the three ISO20022 codes:	String	OP	**ExternalBankTransactionDomain1Code** • Ex: "bankTransactionCode": "PMNT-ICDT-STDO"

	• Domain • Family • Sub-family Separated by "-" Example: PMNT-RCTD-ESCT defines a transaction assigned to the PayMeNT domain (PMNT), ReceivedCreDitTransfer family (RCDT) and EuropeanSEPACreditTransfer (ESCT).			
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: • transactionDetails	Links	OP	Ex: "_links": {…}

8.21 TrustedBeneficiary

Field	Description	Type	Mandat.	Format
trustedBeneficiaryId	Resource identifier of the entry in the list.	String	OB	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	OB	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	OB	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	OB	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	OB	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.
Header	String	OP	It is used to specify the list of HTTP headers included when the signature for the message is generated. 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.	The mandatory fields to sign are: • digest • x-request-id Conditionally, if they travel and are supported, it can include: • psu-id • psu-corporate-id • tpp-redirect-uri

			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.	
Signature	String	OB	The "signature" parameter must be in Base64 ACCORDING to RFC 4648. 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.	There are no additional requirements.

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: A5F1CF405B28E44ED29507E0F64495859BA877893D2A714512D16CE3BD8BE562
- Base64: pfHPQFso5E7SlQfg9kSVhZuod4k9KnFFEtFs472L5WI=

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

SHA256=pfHPQFso5E7SlQfg9kSVhZuod4k9KnFFEtFs472L5WI=

The headers that we have so far are:

X-Request-ID=a13cbf11-b053-4908-bd06-517dfa3a1861f]
Digest=SHA256=pfHPQFso5E7SlQfg9kSVhZuod4k9KnFFEtFs472L5WI=

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=pfHPQFso5E7SlQfg9kSVhZuod4k9KnFFEtFs472L5WI=

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+5ZVNESdiqKG8xraw
Ya5gAm46CvcKChNTPaakiEJHcXM5RZPWN0Ns5HjV5mUY2QzD+g5mwqcWvXtBr1vg0bZKN
8Zt3+uJMN37NQg9tJNE2yKIJlEPlAYOjC2PA/yzGSLOdADnXQut9yRvxw8gMCjDtRaKDyWmw
G6/crX293hGvBUeff1xvTluWhQzyfx4J6WG0v1ZmpnWdZ1LF68sToeDGTdu65aVKV2q6qcZzc
m5aPV6+mVHX+21Vr6acxiLZdeYUHYJHrzErUN3KJrmt3w2AL7Dw==

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=pfHPQFso5E7SlQfg9kSVhZuod4k9KnFFEtFs472L5WI=

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+5ZVNESdiqKG8xraw
Ya5gAm46CvcKChNTPaakiEJHcXM5RZPWN0Ns5HjV5mUY2QzD+g5mwqcWvXtBr1vg0bZKN
8Zt3+uJMN37NQg9tJNE2yKIJlEPlAYOjC2PA/yzGSLOdADnXQut9yRvxw8gMCjDtRaKDyWmw
G6/crX293hGvBUeff1xvTluWhQzyfx4J6WG0v1ZmpnWdZ1LF68sToeDGTdu65aVKV2q6qcZzc
m5aPV6+mVHX+21Vr6acxiLZdeYUHYJHrzErUN3KJrmt3w2AL7Dw=="

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	• 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	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.

	Also used in DELETE requests of a payment start where authentication is not necessary.
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	Returned if the resource that was referenced in the path exists but cannot be accessed by the TPP or the PSU. When in doubt if a specific path id is sensitive or not, use this code instead of 403.
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 Unsopported 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.
	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	The format of certain fields in the request is wrong. The fields will be indicated. This applies to the body and header fields. It also applies in cases where these entries refer to non-existent or erroneous data instances.
	400	PARAMETER_NOT_CONSISTENT	Parameters sent by the TPP are not consistent. It only applies to query parameters.
	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.
	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.

	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.
	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
ACCC	AcceptedSettelmentCompleted	The settlement in the beneficiary's account has been completed.
ACCP	AcceptedCustomerProfile	The prior verification of the technical validation was correct. The client profile check was also successful.
ACFC	AcceptedFundsChecked	In addition to the client's profile, the availability of funds has been positively verified. Note: needs ISO 20022 approval
ACSC	AcceptedSettlementCompleted	The settlement in the issuer's account has been completed.

		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. 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.
ACSP	AcceptedSettlementInProcess	The previous controls such as technical validations and client profile were correct and, therefore, the payment initiation has been accepted for execution.
ACTC	AcceptedTechnicalValidation	Authentication and syntactic and semantic validation are correct.
ACWC	AcceptedWithChange	The instruction has been accepted, but needs a change, for example, date or other data not sent. Also to inform that a change has been applied, for example, on the start of payment and that the execution date has been changed.
ACWP	AcceptedWithoutPosting	The payment instruction included in the credit transfer has been accepted without being sent to the beneficiary client's account.
RCVD	Received	The payment initiation has been received by the agent (the ASPSP through the HUB)
PATC	PartiallyAcceptedTechnicalCorrect	Payment starts that have been authorized by at least one USP, but have not yet been finally authorized by all applicable USPs. (Multilevel SCA) Note: needs ISO 20022 approval
PDNG	Pending	The payment initiation or individual transaction included in the payment initiation is pending. Additional checks and states updates will be performed.
RJCT	Rejected	The payment initiation or the individual transaction included in the payment start has been rejected.
CANC	Cancelled	The start of payment has been canceled before its execution. Note: needs ISO 20022 approval
PART		A number of transactions were accepted, while another number of transactions have not yet reached the "accepted" state

		Note: this code should be used only in case of Bulk payments. It is only used in situations where all the requested authorizations have been applied, but some payments have been rejected.

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.
partiallyAuthorised	Due to a multilevel SCA, some, but not all of the necessary authorizations have been made.
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 Authentication types

Code	Description
SMS_OTP	SCA method where an OTP associated with the transaction to be authorized is sent to the PSU over an SMS channel.
CHIP_OTP	SCA method where an OTP is generated by an electronic card. To use it, the PSU usually needs a device. The device, after completing the challenge, derives an OTP and shows it to the PSU.
PHOTO_OTP	SCA method where the challenge is a QR or similarly encoded visual data which can be read by a client device or a specific mobile application. The device or application derives a visual challenge OTP and displays it to the PSU.

PUSH_OTP	OTP sent via PUSH to a dedicated authentication APP and displayed to the PSU.
SMTP_OTP	OTP sent via email to the PSU.

9.7 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.
expected	Transactions made up of the entries noted and the entries pending at the time of the request.
openingBooked	Account balance at the beginning of the reporting period. It is always equal to the “closingBooked” balance of the previous period's report.
interimAvailable	Balance available provisionally. Calculated based on the annotations of credit and debit items during the specified period of time.
interimBooked	Balance calculated over the course of the business day, at the specified time and subject to change during the day. This balance is calculated taking the credit and debit items noted during the specified time / period.
forwardAvailable	Advance of the balance of available cash that is available to the account holder on the specified date.

9.8 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

9.9 SCA states

Code	Description
received	The authorization resource has been created successfully.
psuIdentified	The PSU associated with the authorization resource has been identified and authenticated, for example, by a password or by the access token
psuAuthenticated	The PSU related to the authorization or cancellation authorization resource has been identified and authenticated, for example, by password or access token.
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.
unconfirmed	The SCA has been technically successfully terminated by the PSU, but the authorization resource needs the confirmation request from the TPP.
finalised	The SCA flow has been completed successfully.
failed	SCA flow has failed.
exempted	The transaction is exempt from SCA, the associated authorization is correct.

9.10 Guide of good practice

9.10.1 Campo remittanceInformationUnstructured

This field can be used following the EACT standard *" Association of European Treasurers"* and adopted in BG in *" Mobile P2P Interoperability Framework - Implementation Guidelines v1.0"*

The format is as follows:

Field	Description
/ DNR /	Issuer alias
/ CNR /	Beneficiary alias. (Recommended to send FUC from the merchant)
/ DOC /	Reference data of the corresponding request. (El Hub monta X-Request-Id del TPP)

/ TXT /	Additional text / concept

Example
"remittanceInformationUnstructured": "/DOC/db617660-d60d-11e8-9f8b-f2801f1b9fd1/TXT/Compra en comercio xxx"

9.10.2 Lifetime of the scaRedirect link

Berlin Group recommends a duration of 5 minutes for this type of link.