Payment Initiation API Specification - v3.1.2

Version Control

Version	Date	Author	Comments
3.1	03 Dec 2018	OB R/W API Team	This is the baseline version.
4.0-draft1	13 Dec 2018	OB R/W API Team	v4.0-draft1 changes: • Clarified a PISP can optionally confirm available funds for an international scheduled (immediate debit) payments in Overview section. • Swagger links updated.
4.0-draft3	15 Jan 2019	OB R/W API Team	v4.0-draft3 changes: • Added scope description • Added note that PIS Specification can be used with Trusted Beneficiaries Specification
4.0-draft4	04 Feb 2019	OB R/W API Team	v4.0-draft4 changes: • Removed references of Trusted Beneficiary Specification
4.0-draft6	04 Mar 2019	OB R/W API Team	v4.0-draft6 changes: • Added AcceptedWithoutPosting and AcceptedCreditSettlementCompleted in OBTransactionIndividualStatus1Code enumeration list. • Added OBTransactionIndividualExtendedISOStatus1Code, OBTransactionIndividualStatusReason1Code and OBExternalStatus3Code enumeration lists • Added reusable class OBWritePaymentDetails1 which provides payment-details, for all Payment Orders.
4.0-draft7	08 Mar 2019	OB R/W API Team	v4.0-draft7 changes: • Updated the Flow diagram step 6 to reflect the order of consent, payment-order and payment-details resource call • Added new static enumerations • Added reusable class OBSCASupportData1, which is used with all Payment Order Consent resources to enable PISPs to provide Supporting Data, when requesting SCA Exemption
3.1.2-RC1	17 Apr 2019	OB R/W API Team	Renamed to 3.1.2-RC1

Overview

This Payment Initiation API Specification describes the flows and payloads for initiating a **general** payment-order.

The API endpoints described here allow a PISP to:

- Register an intent to **stage** a payment-order consent.
- Optionally confirm available funds for a payment-order
    - Domestic immediate, international immediate and international scheduled (immediate debit) payments only.
- Subsequently **submit** the payment-order for processing.
- Optionally retrieve the status of a payment-order **consent** or payment-order **resource.**

This specification should be read in conjunction with Read/Write Data API Specification which provides a description of the elements that are common across all the Read/Write Data APIs.

Document Overview

This document consists of the following parts:

Overview: Provides an overview of the API and the key decisions and principles that contributed to the specification.

Basics: The section begins with an introduction to how the API is to be used to initiate a payment order, using the example of a single immediate payment. It goes on to identify the resources and operations that are permitted on those resources and various special cases.

Security & Access Control: Specifies the means for PISPs and PSUs to authenticate themselves and provide consent.

Swagger Specifications: Provides links to the swagger specifications for the APIs.

Data Model: Describes the data model for the API payloads.

Usage Examples: Provides examples for normal flows, and alternate flows.

Design Principles

Scheme Agnostic

The API has been be designed so that it is agnostic to the underlying payment scheme that is responsible for carrying out the payment.

In doing so, this means we will not design field lengths and payloads to only match the Faster Payments message, and will instead rely on the field lengths and definitions in ISO 20022. Due diligence has been carried out to ensure that the API has the necessary fields to function with Bacs payments as per the agreed scope.

Further mapping guidance has been provided to ensure that differences are understood between the Open Banking Payment API standard, and other message formats in the Domestic Payment Message Formats sub-page.

Status Codes

The API uses two status codes that serve two different purposes:

- The HTTP Status Code reflects the outcome of the API call (the HTTP operation on the resource).
- The Status field for the payment-order consent reflects the status of the PSU consent authorisation.
- The Status field for the payment-order resource reflects the status of the payment-order initiation or execution.

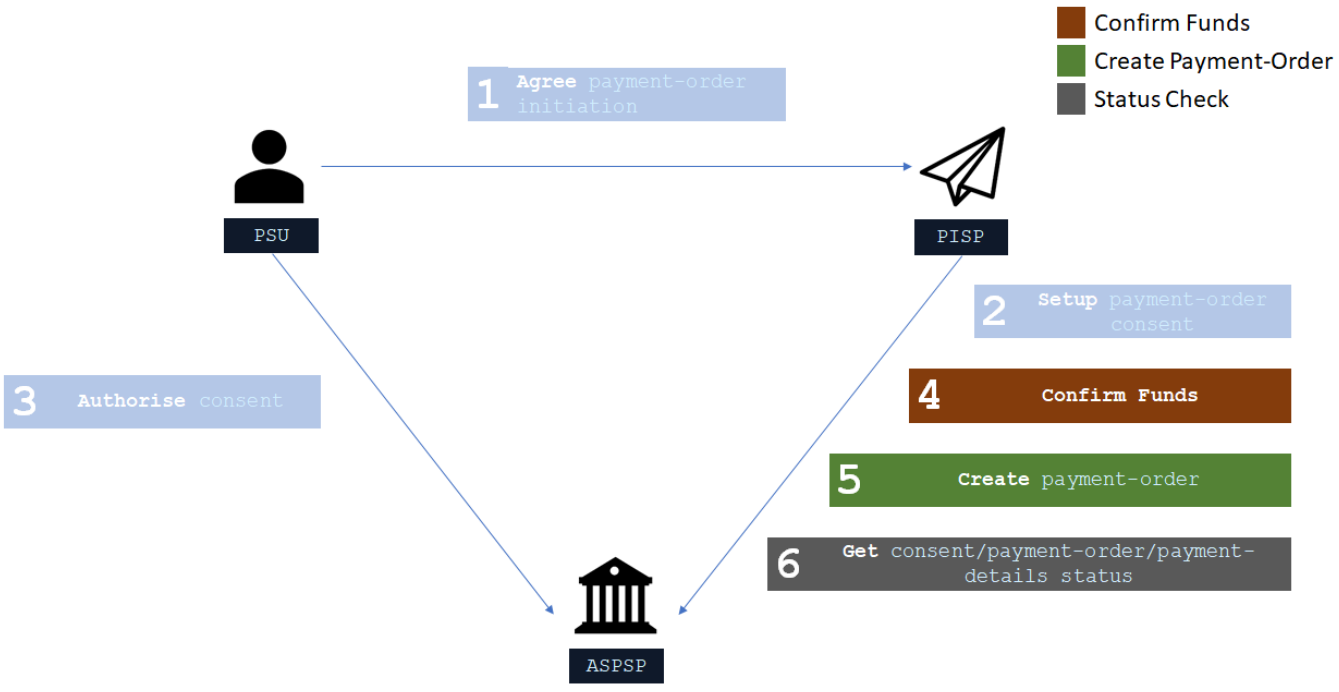
Basics

Overview

The figure below provides a **general** outline of a payment flow for all payment-order types using the Payment APIs. The payment-order types covered in this specification include:

- Domestic payments.
- Domestic scheduled payments.
- Domestic standing orders.
- International payments.
- International scheduled payments.

The payment-order consent and payment-order resource in the following flow generalises for the different payment-order types. e.g. for a domestic payment, the payment-order consent resource is domestic-payment-consents; and the payment-order resource is domestic-payments.



Steps

Step 1: Agree Payment-Order Initiation

- This flow begins with a PSU consenting to a payment being made. The consent is between the PSU and the PISP.
- The debtor account details can optionally be specified at this stage.

Step 2: Setup Payment-Order Consent

- The PISP connects to the ASPSP that services the PSU's payment account and creates a new **payment-order consent** resource. This informs the ASPSP that one of its PSUs intends to make a **payment-order**. The ASPSP responds with an identifier for the payment-order consent resource (the ConsentId, which is the intent identifier).
- This step is carried out by making a **POST** request to the **payment-order consent** resource.

Step 3: Authorise Consent

- The PISP requests the PSU to authorise the consent. The ASPSP may carry this out by using a *redirection flow* or a *decoupled flow*.
    - In a redirection flow, the PISP redirects the PSU to the ASPSP.
        - The redirect includes the ConsentId generated in the previous step.
        - This allows the ASPSP to correlate the payment order consent that was setup.
        - The ASPSP authenticates the PSU.
        - The PSU selects the debtor account at this stage (if it has not been previously specified in Step 1).
        - The ASPSP updates the state of the payment order consent resource internally to indicate that the consent has been authorised.
        - Once the consent has been authorised, the PSU is redirected back to the PISP.
    - In a decoupled flow, the ASPSP requests the PSU to authorise consent on an *authentication device* that is separate from the *con sumption device* on which the PSU is interacting with the PISP.
        - The decoupled flow is initiated by the PISP calling a back-channel authorisation request.
        - The request contains a 'hint' that identifies the PSU paired with the consent to be authorised.
        - The ASPSP authenticates the PSU
        - The PSU selects the debtor account at this stage (if it has not been previously specified in Step 1)
        - The ASPSP updates the state of the payment order consent resource internally to indicate that the consent has been authorised.
        - Once the consent has been authorised, the ASPSP can make a callback to the PISP to provide an access token.

Step 4: Confirm Funds (Domestic and International Single Immediate Payments Only)

- Once the PSU is authenticated and authorised the **payment-order-consent**, the PISP can check whether funds are available to make the payment.
- This is carried out by making a **GET** request, calling the **funds-confirmation** operator on the **payment-order-consent** resource.

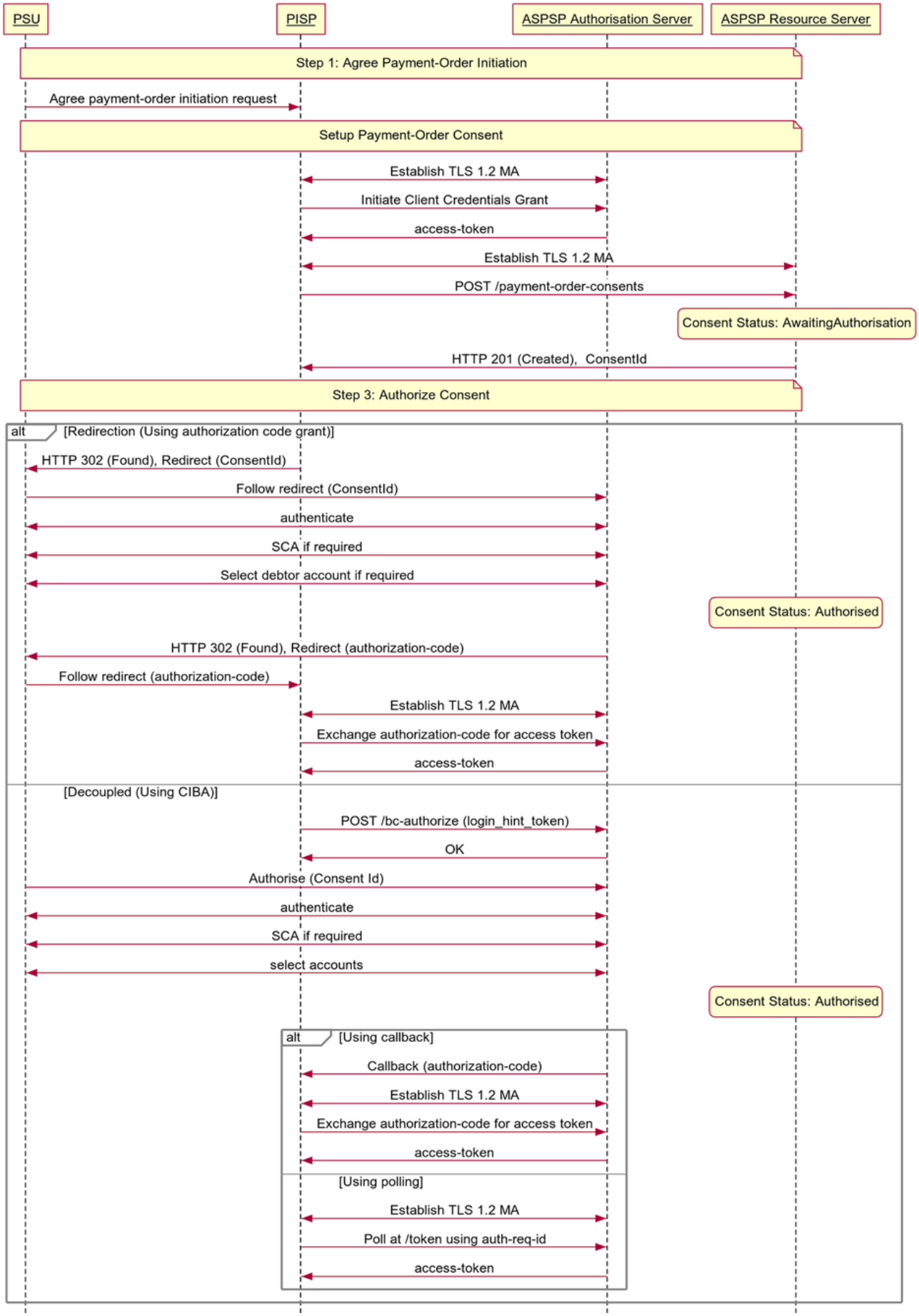
Step 5: Create Payment-Order

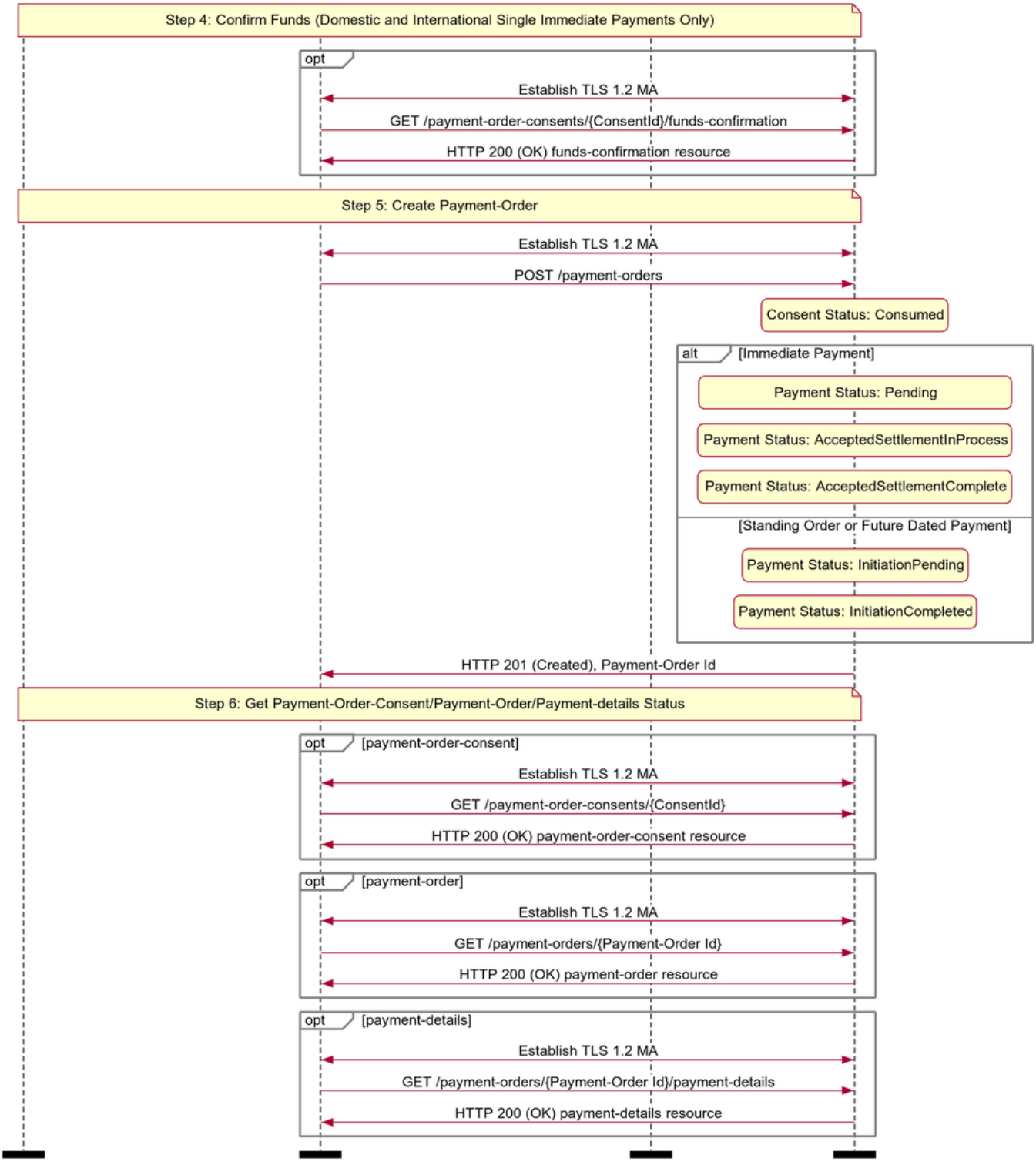
- The PISP creates a **payment-order** resource to indicate that the payment created in the steps above should be submitted for processing.
- This is carried out by making a **POST** request to the appropriate **payment-order** resource.
- The ASPSP returns the identifier for the payment-order resource to the PISP.

Step 6: Get Consent/Payment-Order/Payment-Details Status

- The PISP can check the status of the payment-order consent (with the ConsentId) or payment-order resource (with the payment-order resource identifier) or payment-details(with the payment-order resource identifier) .
- This is carried out by making a **GET** request to the **payment-order consent** or **payment-order** or **payment-details** resource.

Sequence Diagram


PSU
PISP
ASPSP Authorisation Server
ASPSP Resource Server
Step 1: Agree Payment-Order Initiation
Agree payment-order initiation request
Setup Payment-Order Consent
Establish TLS 1.2 MA
Initiate Client Credentials Grant
access-token
Establish TLS 1.2 MA
POST /payment-order-consents
Consent Status: AwaitingAuthorisation
HTTP 201 (Created), ConsentId
Step 3: Authorize Consent
alt
[Redirection (Using authorization code grant)]
HTTP 302 (Found), Redirect (ConsentId)
Follow redirect (ConsentId)
authenticate
SCA if required
Select debtor account if required
Consent Status: Authorised
HTTP 302 (Found), Redirect (authorization-code)
Follow redirect (authorization-code)
Establish TLS 1.2 MA
Exchange authorization-code for access token
access-token
[Decoupled (Using CIBA)]
POST /bc-authorize (login_hint_token)
OK
Authorise (Consent Id)
authenticate
SCA if required
select accounts
Consent Status: Authorised
alt
[Using callback]
Callback (authorization-code)
Establish TLS 1.2 MA
Exchange authorization-code for access token
access-token
[Using polling]
Establish TLS 1.2 MA
Poll at /token using auth-req-id
access-token



Payment Initiation - High Level Flow

Payment Restrictions

The standard does not provide a uniform set of restrictions for payment-order types that can be supported through this API.

For example, but not limited to:

- The maximum InstructedAmount allowable.
- The domestic-standing-order Frequency patterns supported.
- The maximum future date on a scheduled-payment.

Each ASPSP **must** determine appropriate restrictions that they support based on their individual practices, standards and limitations. These restrictions should be documented on ASPSP developer portals.

An ASPSP **must** reject the payment-order **consent** if the ASPSP is unable to handle the request.

CutOffDateTime Behaviour

An ASPSP **may** return the specific CutOffDateTime when responding to a payment-order consent request.

An ASPSP **must** document the behaviour for a payment receipt before and after the CutOffDateTime for a payment-order has elapsed.

Two strategies for handling behaviour are:

- Reject the payment-order (and steps associated with the creation of payment-order) if received after the applicable CutOffDateTime
- Accept the payment-order (and steps associated with the creation of payment-order) if received after the applicable CutOffDateTime

Reject the Payment-Order

In this scenario, the behaviour of payment-order execution is explicit to the PISP and PSU.

- An ASPSP **must** reject the payment-order **consent** if the CutOffDateTime for a specific payment-order type has elapsed.
- An ASPSP **must** reject an authorization request when the underlying intent object is associated with a CutoffDateTime that has elapsed. The ASPSP **must not** issue an access token in such a situation. The ASPSP **must** set the status of the payment-order consent resource to “Rejected”.
- An ASPSP **must** reject the payment-order **resource** if the CutOffDateTime for a specific payment-order type, has been established and has elapsed.
- A PISP **must** ensure that the PSU consent authorisation is completed and the payment-order resource is created before the CutOffDateTime elapses.

For a payment-order **consent** or a payment-order **resource** that has been rejected due to the elapsed CutoffDateTime, the PISP **may** decide to create a corresponding schedule payment endpoint to create a new payment-order consent. E.g. if a PISP attempts to make a BACS payment after 16:00, it would be rejected. The PISP may use the /domestic-scheduled-payment-consents endpoint to create a consent for the same payment for the next working day.

Accept the Payment-Order

In this scenario, the behaviour of the payment-order execution is not explicit to the PISP and PSU, and the payment-order will be executed on the next available working day.

- An ASPSP **must** accept the payment-order **consent** if the CutOffDateTime for a specific payment-order type has elapsed.
- An ASPSP **must** accept an authorization request when the underlying intent object is associated with a CutoffDateTime that has elapsed.
- An ASPSP **must** accept the payment-order **resource** if the CutOffDateTime for a specific payment-order type, has been established and has elapsed.
- An ASPSP **may** update the payment-order consent or payment-order **resource** with the CutOffDateTime, ExpectedExecutionDateTime and ExpectedSettlementDateTime, to communicate expected execution behaviour if the CutOffDateTime has elapsed.

Release Management

This section overviews the release management and versioning strategy for the Payment Initiation API. It applies to all Payment Order Consent and Payment Order resources, specified in the Endpoints section.

Payment-Order Consent

POST

- A PISP **must not** create a payment-order consent ConsentId on a newer version and use it to create a payment-order resource in a previous version
    - E.g., A ConsentId created in v3, must not be used to create a v1 PaymentSubmissionId
- A PISP **must not** create a payment-order consent ConsentId on a previous version and use it to create a payment-order resource in a newer version
    - E.g., A PaymentId created in v1, must not be used to create a v3 DomesticPaymentId

GET

- A PISP **must not** access a payment-order ConsentId created in a newer version, via a previous version endpoint
    - E.g., A ConsentId created in v3 accessed via a v1 PaymentId
- An ASPSP **may** choose to make ConsentIds accessible across versions
    - E.g., for a PaymentId created in v1, an ASPSP may or may not make it available via v3, as this is a short-lived consent

Payment-Order Consent (Confirm Funds)

GET

- A PISP must not confirm funds using a payment-order-consent ConsentId created in a different version.
    - E.g. A ConsentId created in v3, must not be used to confirm funds on a v1 endpoint.

Payment-Order Resource

POST

- A PISP **must** use a payment-order consent ConsentId within the same version to create the payment-order resource (in that version)
    - E.g., A v3 payment-order **consent** can only be used to create a payment-order **resource** in v3.
- An ASPSP **must not** allow a PISP to use a ConsentId from a previous version to create a Payment Order in a newer version, and vice versa

GET

- A PISP **must** refer to the ASPSP's online Developer Portal for guidelines on accessibility of a payment-order resource in a newer version
- A PISP **must** not access the payment-order resource types introduced in a newer version, on an older version endpoint:
    - E.g., an international-payment created in v3, that is accessed via the v1 payment-submissions endpoint.
- A PISP **must** not access the payment-order resource created in a newer version on an older version endpoint:
    - E.g., for a domestic-payment resource created in v3, access via the v1 payment-submissions endpoint is not permitted.
- An ASPSP **must** document the behaviour on the accessibility of a payment-order resource in a newer version on the ASPSP's online Developer Portal.
- An ASPSP **must** allow access to the payment-order resource created in a previous version on a newer version endpoint (depending on an ASPSP's legal requirement for data retention):
    - E.g., a payment-submission created in v1, must be accessible as a v3 domestic-payment, with sensible defaults for additional fields introduced in v3 (e.g., if an ASPSP must make payment resources available for 7 years).
    - In the case where a payment-order type is the same, but the structure has changed in a newer version, sensible defaults may be used, with the ASPSP's Developer Portal clearly specifying the behaviour.
        - E.g., a new field StatusUpdateDateTime was introduced in v3, an ASPSPs must populate this with the last status update time (as the StatusUpdateDateTime is a mandatory field).

Endpoints

This section looks at the list of available API endpoints to complete a Payment flow and optionality (definitions of mandatory, conditional or optional are defined in the Design Principles section in Read/Write Data API Specification). For detail on the request and response objects, please refer to the Data Model section of the specification.

The Mandatory/Conditional/Optional status of a resource's POST endpoint matches the GET operation. If a POST endpoint is implemented, the GET endpoint **must** also be implemented.

Endpoint design considerations:

- Having a separate resource for the payment-order consent and payment-order resource means we can extend the flows in the future.
- Separation in the payment-order consent and the payment-order resource also allows for cleaner separation in updating the status of resources for ASPSPs that chose to implement the functionally.

Link	Resource	Endpoints
Domestic Payments v3.1.2	domestic-payment-consents	POST /domestic-payment-consents
	domestic-payment-consents	GET /domestic-payment-consents/{ConsentId}
	domestic-payment-consents	GET /domestic-payment-consents/{ConsentId}/funds-confirmation
	domestic-payments	POST /domestic-payments
	domestic-payments	GET /domestic-payments/{DomesticPaymentId}

	payment-details	GET /domestic-payments/{DomesticPaymentId}/payment-details
Domestic Scheduled Payment v3.1.2	domestic-scheduled-payment-consents	POST /domestic-scheduled-payment-consents
	domestic-scheduled-payment-consents	GET /domestic-scheduled-payment-consents/{ConsentId}
	domestic-scheduled-payments	POST /domestic-scheduled-payments
	domestic-scheduled-payments	GET /domestic-scheduled-payments/{DomesticScheduledPaymentId}
	payment-details	GET /domestic-scheduled-payments/{DomesticScheduledPaymentId}/payment-d
Domestic Standing Orders v3.1.2	domestic-standing-order-consents	POST /domestic-standing-order-consents
	domestic-standing-order-consents	GET /domestic-standing-order-consents/{ConsentId}
	domestic-standing-orders	POST /domestic-standing-orders
	domestic-standing-orders	GET /domestic-standing-orders/{DomesticStandingOrderId}
	payment-details	GET /domestic-standing-orders/{DomesticStandingOrderId}/payment-details
International Payments v3.1.2	international-payment-consents	POST /international-payment-consents
	international-payment-consents	GET /international-payment-consents/{ConsentId}
	international-payment-consents	GET /international-payment-consents/{ConsentId}/funds-confirmation
	international-payments	POST /international-payments
	international-payments	GET /international-payments/{InternationalPaymentId}
	payment-details	GET /international-payments/{InternationalPaymentId}/payment-details
International Scheduled Payments v3.1.2	international-scheduled-payment-consents	POST /international-scheduled-payment-consents
	international-scheduled-payment-consents	GET /international-scheduled-payment-consents/{ConsentId}
	international-scheduled-payment-consents	GET /international-scheduled-payment-consents/{ConsentId}/funds-confirmation
	international-scheduled-payments	POST /international-scheduled-payments
	international-scheduled-payments	GET /international-scheduled-payments/{InternationalScheduledPaymentId}
	payment-details	GET /international-scheduled-payments/{InternationalScheduledPaymentId}/payment-

International Standing Orders v3.1.2	international-standing-order-consents	POST /international-standing-order-consents
	international-standing-order-consents	GET /international-standing-order-consents/{ConsentId}
	international-standing-orders	POST /international-standing-orders
	international-standing-orders	GET /international-standing-orders/{InternationalStandingOrderPaymentId}
	payment-details	GET /international-standing-orders/{InternationalStandingOrderPaymentId}/payment-c
File Payments v3.1.2	file-payment-consents	POST /file-payment-consents
	file-payment-consents	GET /file-payment-consents/{ConsentId}
	file-payment-consents	POST /file-payment-consents/{ConsentId}/file
	file-payment-consents	GET /file-payment-consents/{ConsentId}/file
	file-payments	POST /file-payments
	file-payments	GET /file-payments/{FilePaymentId}
	file-payments	GET /file-payments/{FilePaymentId}/report-file
	payment-details	GET /file-payments/{FilePaymentId}/payment-details

Security & Access Control

Scopes

The access tokens required for accessing the Payment APIs must have at least the following scope:

Scopes

```
payments: Generic payment scope
```

Grants Types

PISPs **must** use a client credentials grant to obtain a token to make POST requests to the payment-order **consent** endpoints. In the specification, this grant type is referred to as "Client Credentials".

PISPs **must** use an authorization code grant using a redirect or decoupled flow to obtain a token to make POST requests to the payment-order **re source** endpoints. This token may also be used to confirm funds on a payment-order **consent** resource. In the specification, this grant type is referred to as "Authorization Code".

PISPs **must** use a client credentials grant to obtain a token to make GET requests (excluding confirming funds).

Consent Authorisation

OAuth 2.0 scopes are coarse-grained and the set of available scopes are defined at the point of client registration. There is no standard method for specifying and enforcing fine-grained scopes e.g., a scope to enforce payments of a specified amount on a specified date.

A *consent authorisation* is used to define the fine-grained scope that is granted by the PSU to the PISP.

The PISP **must** begin a payment-order request by creating a **payment-order consent** resource through a **POST** operation. These resources indicate the *consent* that the PISP claims it has been given by the PSU. At this stage, the consent is not yet authorised as the ASPSP has not yet verified this claim with the PSU.

The ASPSP responds with a ConsentId. This is the intent-id that is used when initiating the authorization code grant (as described in the Trust Framework).

As part of the authorization code grant:

- The ASPSP authenticates the PSU.
- The ASPSP plays back the consent (registered by the PISP) back to the PSU to get consent authorisation. The PSU may accept or reject the consent in its entirety (but not selectively).
- If the consent did not indicate a debtor account the ASPSP presents the PSU with a list of accounts from which the PSU may select one.

Once these steps are complete, the consent is considered to have been authorised by the PSU.

Multiple Authorisation

In a multiple authorisation context, the same consent authorisation steps are followed for the first PSU to authorise or stage the payment-order consent.

In the payment-order consent:

- A PISP **may** request an AuthorisationType for the payment-order (i.e., Single or Any). If a value is not provided, an ASPSP will interpret the AuthorisationType as 'Any'.
- A PISP **may** request a CompletionDateTime for the payment-order authorisation to be complete. If a value is not provided, an ASPSP will interpret the CompletionDateTime as unbounded.
- An ASPSP **must** reject the payment-order consent if the AuthorisationType requested by the PISP does not match the DebtorAccount in the request.
- An ASPSP **must** set the status of the payment-order consent to Rejected, if the AuthorisationType requested by the PISP cannot be satisfied, after PSU Authentication:
    - The ASPSP must respond back with an OAuth error response fields ***error*** specified as invalid_request and ***error_description*** co ntaining an appropriate message.
- An ASPSP **must** restrict the selection of DebtorAccount (in the ASPSP online channel) to accounts that match the AuthorisationType requested by the PISP.

In the payment-order resource:

- An ASPSP **must** respond with the MultiAuthorisation object if the payment-order requires multiple authorisations. The MultiAuthorisation object indicates to the PISP that the payment-order requires multiple authorisations.
- The ASPSP **must** populate the MultiAuthorisation object with the Status of the multiple authorisaitons.
- The ASPSP **may** populate the MultiAuthorisation object with additional details of the multiple authorisation journey including:
    - The number of required authorisations (total required at the start of the multi authorisation journey).
    - The number of authorisations complete.
    - The date and time of the last authorisation update.
    - The date and time that the authorisation flow must be completed.

Once the final authorisation is received by the ASPSP, the ASPSP **may** notify the PISP that the payment-order resource has been fully Authorised using an Event Notification (as described in the Event Notification API Specification).

Error Condition

If the PSU does not complete a successful consent authorisation (e.g., if the PSU has not authenticated successfully), the authorization code grant ends with a redirection to the TPP with an error response as described in RFC 6749 Section 4.1.2.1. The PSU is redirected to the TPP with an error parameter indicating the error that occurred.

Consent Revocation

A PSU cannot revoke a payment-order consent once it has been authorized.

This is required to comply with Article 80 of PSD2.

Changes to Selected Account

For a payment-order consent, the selected debtor account cannot be changed once the consent has been authorized.

Consent Re-authentication

Payment consents are short-lived and cannot be re-authenticated by the PSU.

Risk Scoring Information

During the design workshops, ASPSPs articulated a need to perform risk scoring on the payments initiated via the Payment API.

Information for risk scoring and assessment will come via:

- FAPI HTTP headers. These are defined in Section 6.3 of the FAPI specification and in the Headers section above.
- Additional fields identified by the industry as business logic security concerns which will be passed in the Risk section of the payload in the JSON object.

These are the set of additional fields in the risk section of the payload for v1.0 which will be specified by the PISP:

- PaymentContextCode.
- MerchantCategoryCode.
- MerchantCustomerIdentification.
- DeliveryAddress.

The PaymentContextCode describes the payment context and can have these values:

- BillPayment.
- EcommerceGoods.
- EcommerceServices.
- Other.
- PartyToParty.

Payments for EcommerceGoods and EcommerceServices will be expected to have a MerchantCategoryCode and MerchantCustomerIdentification populated. Payments for EcommerceGoods will also have the DeliveryAddress populated.

These fields are documented further in the Data Payload section.

Swagger Specification

The Swagger Specification for Payment Initiation APIs can be downloaded from the following links:

- JSON
- YAML

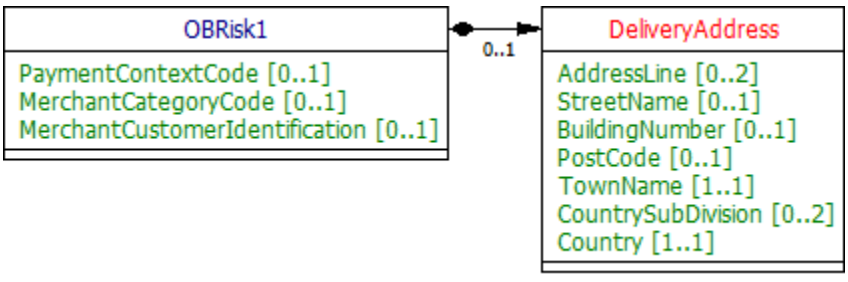
Data Model

Reused Classes

OBRisk1

This section describes the Risk1 class which is reused in the payment-order consent and payment-order resources.

UML Diagram



Data Dictionary

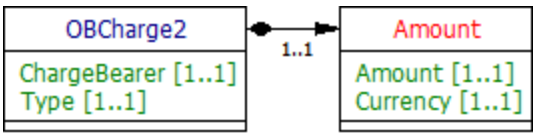
Name	Occurrence	XPath	EnhancedDefinition	Class

OBRisk1		OBRisk1	The Risk section is sent by the initiating party to the ASPSP. It is used to specify additional details for risk scoring for Payments.	OBRisk1
PaymentContextCode	0..1	OBRisk1/PaymentContextCode	Specifies the payment context	OBExternalPaymentContext1Code
MerchantCategoryCode	0..1	OBRisk1/MerchantCategoryCode	Category code conform to ISO 18245, related to the type of services or goods the merchant provides for the transaction.	Min3Max4Text
MerchantCustomerIdentification	0..1	OBRisk1/MerchantCustomerIdentification	The unique customer identifier of the PSU with the merchant.	Max70Text
DeliveryAddress	0..1	OBRisk1/DeliveryAddress	Information that locates and identifies a specific address, as defined by postal services or in free format text.	PostalAddress18
AddressLine	0..2	OBRisk1/DeliveryAddress/AddressLine	Information that locates and identifies a specific address, as defined by postal services, that is presented in free format text.	Max70Text
StreetName	0..1	OBRisk1/DeliveryAddress/StreetName	Name of a street or thoroughfare.	Max70Text
BuildingNumber	0..1	OBRisk1/DeliveryAddress/BuildingNumber	Number that identifies the position of a building on a street.	Max16Text
PostCode	0..1	OBRisk1/DeliveryAddress/PostCode	Identifier consisting of a group of letters and/or numbers that is added to a postal address to assist the sorting of mail.	Max16Text
TownName	1..1	OBRisk1/DeliveryAddress/TownName	Name of a built-up area, with defined boundaries, and a local government.	Max35Text
CountrySubDivision	0..2	OBRisk1/DeliveryAddress/CountrySubDivision	Identifies a subdivision of a country, for instance state, region, county.	Max35Text
Country	1..1	OBRisk1/DeliveryAddress/Country	Nation with its own government, occupying a particular territory.	CountryCode

OBCharge2

This section describes the OBCharge2 class - which is reused in the response payloads in the payment-order consent and payment-order resources.

UML Diagram



Data Dictionary

Name	Occurrence	XPath	EnhancedDefinition	Class	Codes
OBCharge2		OBCharge2	Set of elements used to provide details of a charge for the payment initiation.	OBCharge2	
ChargeBearer	1..1	OBCharge2/ChargeBearer	Specifies which party/parties will bear the charges associated with the processing of the payment transaction.	OBChargeBearerType1Code	BorneByCreditor BorneByDebtor FollowingServiceLevel Shared
Type	1..1	OBCharge2/Type	Charge type, in a coded form.	OBExternalPaymentChargeType1Code	
Amount	1..1	OBCharge2/Amount	Amount of money associated with the charge type.	OBActiveOrHistoricCurrencyAndAmount	
Amount	1..1	OBCharge2/Amount/Amount	A number of monetary units specified in an active currency where the unit of currency is explicit and compliant with ISO 4217.	OBActiveCurrencyAndAmount_SimpleType	
Currency	1..1	OBCharge2/Amount/Currency	A code allocated to a currency by a Maintenance Agency under an international identification scheme, as described in the latest edition of the international standard ISO 4217 "Codes for the representation of currencies and funds".	ActiveOrHistoricCurrencyCode	

OBAuthorisation1

This section describes the OBAuthorisation1 class which is used in the payment-order consent request and payment-order consent response payloads.

UML Diagram

Data Dictionary

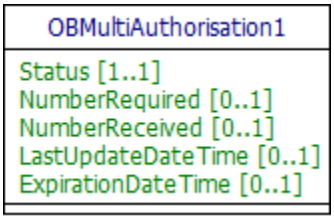
Name	Occurrence	XPath	EnhancedDefinition	Class	Codes	Pattern
OBAuthorisation1		OBAuthorisation1	The authorisation type request from the TPP.	OBAuthorisation1		
AuthorisationType	1..1	OBAuthorisation1/AuthorisationType	Type of authorisation flow requested.	OBExternalAuthorisation1Code	Any Single	

CompletionDateTime	0..1	OBAuthorisation1/CompletionDateTime	Date and time at which the requested authorisation flow must be completed.	ISODateTime		

OBMultiAuthorisation1

This section describes the OBMultiAuthorisation1 class which used in the response payloads of payment-order resources.

UML Diagram



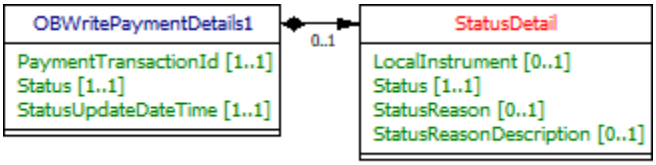
Data Dictionary

Name	Occurrence	XPath	EnhancedDefinition	Class	Codes
OBMultiAuthorisation1		OBMultiAuthorisation1	The multiple authorisation flow response from the ASPSP.	OBMultiAuthorisation1	
Status	1..1	OBMultiAuthorisation1/Status	Specifies the status of the authorisation flow in code form.	OBExternalStatus2Code	Authorised AwaitingFurtherAutho Rejected
NumberRequired	0..1	OBMultiAuthorisation1/NumberRequired	Number of authorisations required for payment order (total required at the start of the multi authorisation journey).	Number	
NumberReceived	0..1	OBMultiAuthorisation1/NumberReceived	Number of authorisations received.	Number	
LastUpdateDateTime	0..1	OBMultiAuthorisation1/LastUpdateDateTime	Last date and time at the authorisation flow was updated.	ISODateTime	
ExpirationDateTime	0..1	OBMultiAuthorisation1/ExpirationDateTime	Date and time at which the requested authorisation flow must be completed.	ISODateTime	

OBWritePaymentDetails1

This section describes the OBWritePaymentDetails1 class which used in the response payloads of payment-detail sub resources.

UML Diagram



Data Dictionary

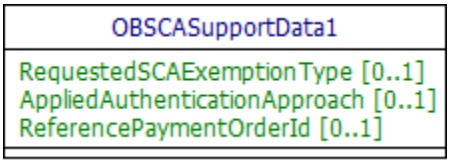
Name	Occurrence	XPath	EnhancedDefinition	Class
OBWritePaymentDetails1	1..1	OBWritePaymentDetails1	Payment status details.	OBWritePaymentDetails1
PaymentTransactionId	1..1	OBWritePaymentDetails1/PaymentTransactionId	Unique identifier for the transaction within an servicing institution. This identifier is both unique and immutable.	Max210Text
Status	1..1	OBWritePaymentDetails1/Status	Status of a transfer, as assigned by the transaction administrator.	OBTransactionIndividualE
StatusUpdateDateTime	1..1	OBWritePaymentDetails1/StatusUpdateDateTime	Date and time at which the status was assigned to the transfer.	ISODateTime
StatusDetail	0..1	OBWritePaymentDetails1/StatusDetail	Payment status details as per underlying Payment Rail.	OBPaymentStatusDetail1
LocalInstrument	0..1	OBWritePaymentDetails1/StatusDetail/LocalInstrument	User community specific instrument. Usage: This element is used to specify a local instrument, local clearing option and/or further qualify the service or service level.	OBExternalLocalInstrume
Status	1..1	OBWritePaymentDetails1/StatusDetail/Status	Status of a transfer, as assigned by the transaction administrator.	Max128Text

StatusReason	0..1	OBWritePaymentDetails1/StatusDetail/StatusReason	Reason Code provided for the status of a transfer.	OBTransactionIndividualS
StatusReasonDescription	0..1	OBWritePaymentDetails1/StatusDetail/StatusReasonDescription	Reason provided for the status of a transfer.	Max256Text

OBSCASupportData1

This section describes the OBSCASupportData1 class, which is used across all *payment order consent* request resources, enabling PISPs to provide Supporting Data when requesting ASPSP for SCA Exemption.

UML Diagram



Data Dictionary

Name	Occurrence	XPath	EnhancedDefinition	Class
OBSCASupportData1		SCASupportData	Supporting Data provided by TPP, when requesting SCA Exemption.	OBSCASupportData1
RequestedSCAExemptionType	0..1	SCASupportData/RequestedSCAExemptionType	This field allows a PISP to request specific SCA Exemption for a Payment Initiation	OBExternalSCAExemptionType1C
AppliedAuthenticationApproach	0..1	SCASupportData/AppliedAuthenticationApproach	Specifies a character string with a maximum length of 40 characters. Usage: This field indicates whether the PSU was subject to SCA performed by the TPP	OBExternalAppliedAuthenticationA

ReferencePaymentOrderId	0..1	SCASupportData/ReferencePaymentOrderId	Specifies a character string with a maximum length of 140 characters. Usage: If the payment is recurring, then the transaction identifier of the previous payment occurrence so that the ASPSP can verify that the PISP, amount and the payee are the same as the previous occurrence.	Max128Text

Identifier Fields

This section describes the identifiers used through the Payment API flows, the direction of flow through the system, and how they are used.

The standard definitions for the elements in the API payloads are described in the Data Payload section. However, this table gives further detail on the business meaning, and how they are used.

Generated	Identifier	Business Description
Merchant/PISP Sent in API Payload	EndToEndIdentification	The EndToEndIdentification reference is a reference that can be populated by the debtor (or merchant in the ecommerce space). This reference is important to the debtor (could be an internal reference Id against the transaction), it Is NOT the reference information that will be primarily populated on the statement of the creditor (beneficiary).
Merchant/PISP Sent in API Payload	InstructionIdentification	The PISP generates the InstructionIdentification which is a unique transaction Id and passes it to the ASPSP (this is mandatory), but this does not have to go any further in the payment flow. The flow of this identifier needs to align with payment scheme rules. The expectation is that this is unique indefinitely across all time periods. The PISP can ensure this is indefinitely unique by including a date or date time element to the field, or by inserting a unique Id.
Merchant/PISP Sent in API Payload	RemittanceInformation	The RemittanceInformation is the reference information that the creditor (or beneficiary) will need to reconcile (e.g. Invoice 123).
ASPSP / API System	ConsentId	A unique identification as assigned by the ASPSP to uniquely identify the payment-order consent resource.
ASPSP / API System	Payment Order Id	Anique identification as assigned by the ASPSP to uniquely identify the payment-order resource. DomesticPaymentId DomesticScheduledPaymentId DomesticStandingOrderId InternationalPaymentId InternationalScheduledPaymentId
ASPSP / Payment Scheme	Scheme Payment ID	This is generated by the ASPSP to uniquely identify a payment through a processing scheme. In the case of FPS, this is the FPID.

The tables below identify the actor that initially creates each of the message identifiers and their transmission and visibility to other actors.

These flows are indicative and will be dependent on what payment schemes or agencies are able to support.

Key:

O indicates the actor that creates the identifier.

=> downstream direction of flow

<= upstream direction of flow

Merchant Flow

Identifier	PSU	Merchant	PISP	ASPSP Originating Bank	Payment Scheme	Beneficiary
EndToEndIdentification		**O**	=>	=>	=>	=>
RemittanceInformation		**O**	=>	=>	=>	=>
InstructionIdentification			**O**	=>		
ConsentId			<=	**O**		
Payment Order Id			<=	**O**		
Scheme Payment ID (e.g., FPID)				**O**	=>	=>

Party to Party Flow

Identifier	PSU	Merchant	PISP	ASPSP Originating Bank	Payment Scheme	Beneficiary
EndToEndIdentification			**O**	=>	=>	=>
RemittanceInformation	**O**		=>	=>	=>	=>
InstructionIdentification			**O**	=>		
ConsentId			<=	**O**		
Payment Order Id			<=	**O**		
Scheme Payment ID (e.g., FPID)				**O**	=>	=>

Payment Order Types

Each of the payment-order types are documented in sub-pages of this specification. Each payment-order type is documented with:

- Endpoints
  - The API endpoints available for the resource.
- Data Model
  - Resource definition.
  - UML diagram.
  - Permissions as they relate to accessing the resource.
  - Data dictionary - which defines fields, re-usable classes, mandatory (1..1) or conditional (0..1) as defined in the Design Principles section, and enumerations.
- Usage Examples.

Enumerations

Static Enumerations

This section gives the definitions for enumerations used in the Payment APIs.

Code Class	Name	Definition
OBExternalPaymentContext1Code	BillPayment	The context of the payment initiation is a bill payment.

OBExternalPaymentContext1Code	EcommerceGoods	The context of the payment initiation is for goods via an ecommerce channel.
OBExternalPaymentContext1Code	EcommerceServices	The context of the payment initiation is for services via an ecommerce channel.
OBExternalPaymentContext1Code	PartyToParty	The context of the payment initiation is a party to party payment.
OBExternalPaymentContext1Code	Other	The context of the payment initiation is of an other type.
OBTransactionIndividualStatus1Code	AcceptedSettlementCompleted	Settlement on the debtor's account has been completed. Usage : this can be used by the first agent to report to the debtor that the transaction has been completed. Warning : this status is provided for transaction status reasons, not for financial information. It can only be used after bilateral agreement. PISPs **must not** use this status as confirmation that settlement is complete on the creditor's account.
OBTransactionIndividualStatus1Code	AcceptedSettlementInProcess	All preceding checks such as technical validation and customer profile were successful and therefore the payment initiation has been accepted for execution.
OBTransactionIndividualStatus1Code	Pending	Payment initiation or individual transaction included in the payment initiation is pending. Further checks and status update will be performed.
OBTransactionIndividualStatus1Code	Rejected	Payment initiation or individual transaction included in the payment initiation has been rejected.
OBTransactionIndividualStatus1Code	AcceptedWithoutPosting	Payment instruction included in the credit transfer is accepted without being posted to the creditor customer's account.
OBTransactionIndividualStatus1Code	AcceptedCreditSettlementCompleted	Settlement on the creditor's account has been completed.
OBExternalConsentStatus1Code	AwaitingAuthorisation	The consent resource is awaiting PSU authorisation.
OBExternalConsentStatus1Code	Rejected	The consent resource has been rejected.
OBExternalConsentStatus1Code	Authorised	The consent resource has been successfully authorised.
OBExternalConsentStatus1Code	Consumed	The consented action has been successfully completed. This does not reflect the status of the consented action.
OBChargeBearerType1Code	BorneByCreditor	All transaction charges are to be borne by the creditor.
OBChargeBearerType1Code	BorneByDebtor	All transaction charges are to be borne by the debtor.
OBChargeBearerType1Code	FollowingServiceLevel	Charges are to be applied following the rules agreed in the service level and/or scheme.

OBChargeBearerType1Code	Shared	In a credit transfer context, means that transaction charges on the sender side are to be borne by the debtor, transaction charges on the receiver side are to be borne by the creditor. In a direct debit context, means that transaction charges on the sender side are to be borne by the creditor, transaction charges on the receiver side are to be borne by the debtor.
OBExternalAuthorisation1Code	Any	Any authorisation type is requested.
OBExternalAuthorisation1Code	Multiple	Multiple authorisation type is requested.
OBExternalAuthorisation1Code	Single	Single authorisation type is requested.
OBExternalStatus1Code	InitiationCompleted	The payment-order initiation has been completed.
OBExternalStatus1Code	InitiationFailed	The payment-order initiation has failed.
OBExternalStatus1Code	InitiationPending	The payment-order initiation is pending.
OBExternalStatus2Code	Authorised	The multiple authorisation flow has been fully authorised.
OBExternalStatus2Code	AwaitingFurtherAuthorisation	The multiple authorisation flow is awaiting further authorisation.
OBExternalStatus2Code	Rejected	The multiple authorisation flow has been rejected.
OBExternalStatus3Code	InitiationCompleted	The payment-order initiation has been completed.
OBExternalStatus3Code	InitiationFailed	The payment-order initiation has failed.
OBExternalStatus3Code	InitiationPending	The payment-order initiation is pending.
OBExternalStatus3Code	Cancelled	Payment initiation has been successfully cancelled after having received a request for cancellation.
OBExchangeRateType2Code	Actual	Exchange rate is the actual rate.
OBExchangeRateType2Code	Agreed	Exchange rate is the agreed rate between the parties.
OBExchangeRateType2Code	Indicative	Exchange rate is the indicative rate.
OBPriority2Code	Normal	Priority is normal.
OBPriority2Code	Urgent	Priority is urgent.
OBAddressTypeCode	Business	Address is the business address.
OBAddressTypeCode	Correspondence	Address is the address where correspondence is sent.
OBAddressTypeCode	DeliveryTo	Address is the address to which delivery is to take place.
OBAddressTypeCode	MailTo	Address is the address to which mail is sent.
OBAddressTypeCode	POBox	Address is a postal office (PO) box.
OBAddressTypeCode	Postal	Address is the complete postal address.
OBAddressTypeCode	Residential	Address is the home address.

OBAddressTypeCode	Statement	Address is the address where statements are sent.
OBTransactionIndividualExtendedISOStatus1Code	Accepted	Request is accepted.
OBTransactionIndividualExtendedISOStatus1Code	AcceptedCancellationRequest	Cancellation is accepted.
OBTransactionIndividualExtendedISOStatus1Code	AcceptedCreditSettlementCompleted	Settlement on the creditor's account has been completed.
OBTransactionIndividualExtendedISOStatus1Code	AcceptedCustomerProfile	Preceding check of technical validation was successful. Customer profile check was also successful.
OBTransactionIndividualExtendedISOStatus1Code	AcceptedFundsChecked	Preceding check of technical validation and customer profile was successful and an automatic funds check was positive.
OBTransactionIndividualExtendedISOStatus1Code	AcceptedSettlementCompleted	Settlement on the debtor's account has been completed. Usage : this can be used by the first agent to report to the debtor that the transaction has been completed. Warning : this status is provided for transaction status reasons, not for financial information. It can only be used after bilateral agreement
OBTransactionIndividualExtendedISOStatus1Code	AcceptedSettlementInProcess	All preceding checks such as technical validation and customer profile were successful and therefore the payment initiation has been accepted for execution.
OBTransactionIndividualExtendedISOStatus1Code	AcceptedTechnicalValidation	Authentication and syntactical and semantical validation are successful
OBTransactionIndividualExtendedISOStatus1Code	AcceptedWithChange	Instruction is accepted but a change will be made, such as date or remittance not sent.
OBTransactionIndividualExtendedISOStatus1Code	AcceptedWithoutPosting	Payment instruction included in the credit transfer is accepted without being posted to the creditor customer’s account.
OBTransactionIndividualExtendedISOStatus1Code	Cancelled	Request is cancelled.
OBTransactionIndividualExtendedISOStatus1Code	NoCancellationProcess	No cancellation process.
OBTransactionIndividualExtendedISOStatus1Code	PartiallyAcceptedCancellationRequest	Cancellation is partially accepted.
OBTransactionIndividualExtendedISOStatus1Code	PartiallyAcceptedTechnicalCorrect	Authentication and syntactical and semantical validation are successful.
OBTransactionIndividualExtendedISOStatus1Code	PaymentCancelled	Transaction has been cancelled.
OBTransactionIndividualExtendedISOStatus1Code	Pending	Payment initiation or individual transaction included in the payment initiation is pending. Further checks and status update will be performed.
OBTransactionIndividualExtendedISOStatus1Code	PendingCancellationRequest	Cancellation request is pending.
OBTransactionIndividualExtendedISOStatus1Code	Received	Payment initiation has been received by the receiving agent.
OBTransactionIndividualExtendedISOStatus1Code	Rejected	Payment initiation or individual transaction included in the payment initiation has been rejected.
OBTransactionIndividualExtendedISOStatus1Code	RejectedCancellationRequest	Cancellation request is rejected

OBTransactionIndividualStatusReason1Code	Cancelled	Reason why the payment status is cancelled
OBTransactionIndividualStatusReason1Code	PendingFailingSettlement	Reason why the payment status is pending (failing settlement).
OBTransactionIndividualStatusReason1Code	PendingSettlement	Reason why the payment status is pending (settlement).
OBTransactionIndividualStatusReason1Code	Proprietary	Defines a free text proprietary reason.
OBTransactionIndividualStatusReason1Code	ProprietaryRejection	Defines the reason that has been used by the Local Instrument system to reject the transaction
OBTransactionIndividualStatusReason1Code	Suspended	Reason why the payment status is suspended.
OBTransactionIndividualStatusReason1Code	Unmatched	Reason why the payment status is unmatched.
OBExternalSCAExemptionType1Code	BillPayment	Bill Payment
OBExternalSCAExemptionType1Code	ContactlessTravel	Contactless Travel
OBExternalSCAExemptionType1Code	EcommerceGoods	Ecommerce Goods
OBExternalSCAExemptionType1Code	EcommerceServices	Ecommerce Services
OBExternalSCAExemptionType1Code	Kiosk	Kisok
OBExternalSCAExemptionType1Code	Parking	Parking
OBExternalSCAExemptionType1Code	PartyToParty	Party To Party
OBExternalAppliedAuthenticationApproach1Code	CA	Single Factor Strong Customer Authentication
OBExternalAppliedAuthenticationApproach1Code	SCA	Multi Factor Strong Customer Authentication

ISO Enumerations

These following ISO Enumerations are used in the Payment APIs.

ISO Data Type	Fields	ISO Enumeration Values URL
Min3Max4Text	MerchantCategoryCode	https://www.iso.org/standard/33365.html
ActiveOrHistoricCurrencyCode	Currency	https://www.iso20022.org/external_code_list.page
CountryCode	Country	https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2#Officially_assigned_code_elements

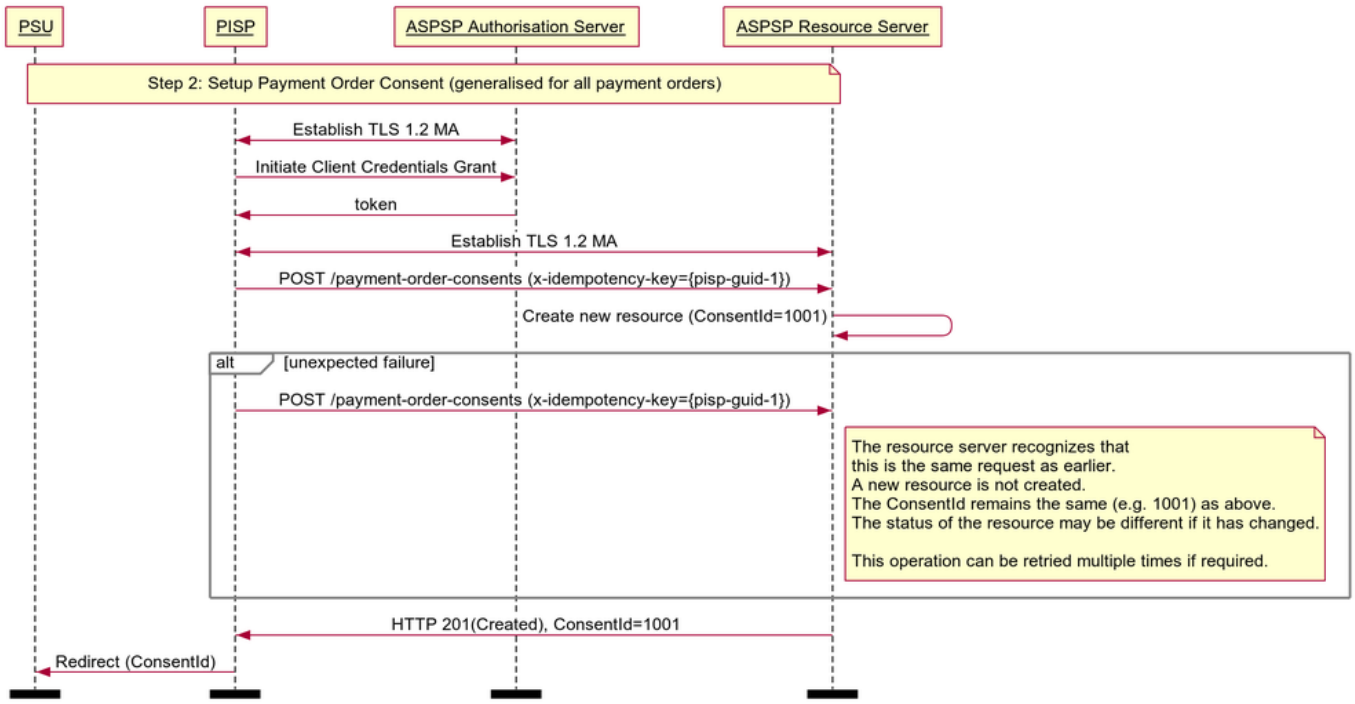
Namespaced Enumerations

The enumerated values specified by Open Banking are documented in Swagger specification and Namespaced Enumerations page.

Alternative and Error Flows

Idempotent Payment Order Consent

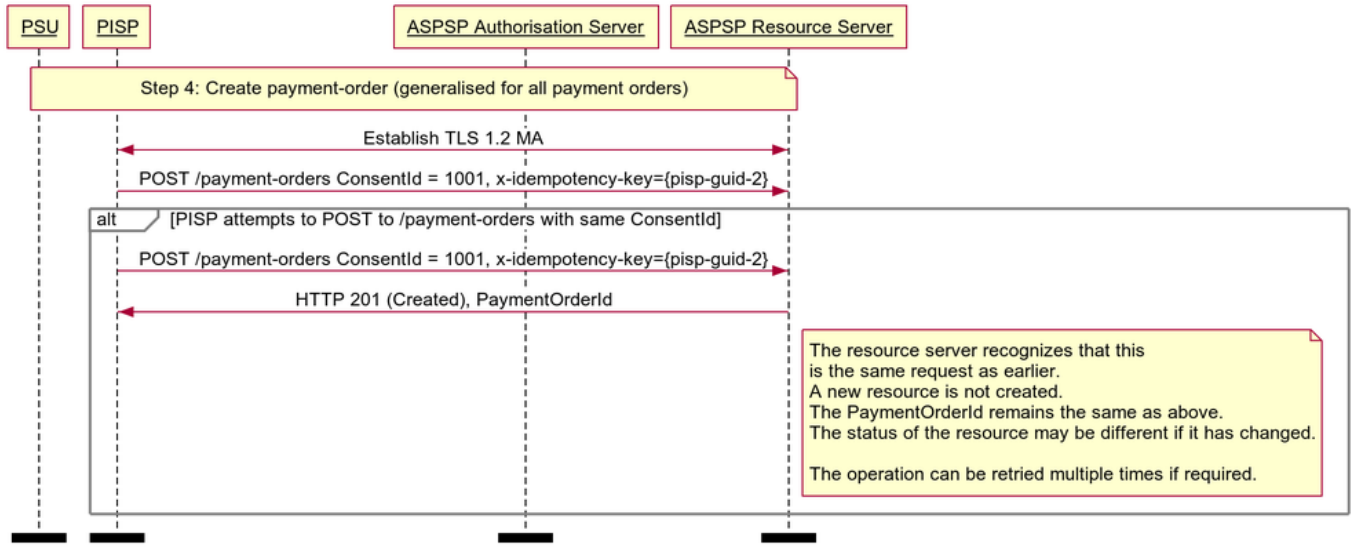
Note: this flow has been generalised for all payment-order types.



Idempotent payment setup

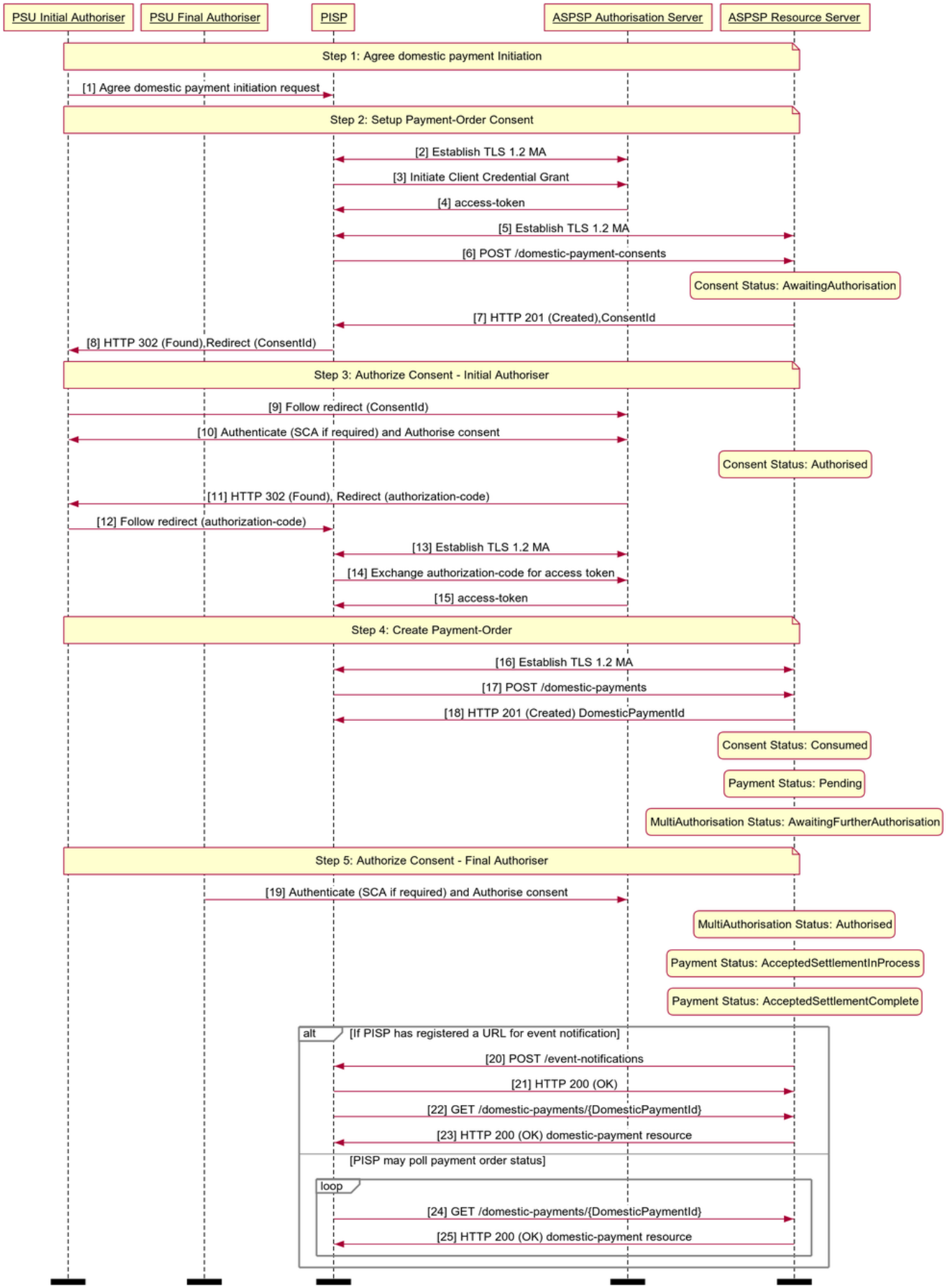
Idempotent Payment Order

Note: this flow has been generalised for all payment-order types.



Payment Order Conflict - Sequence Diagram

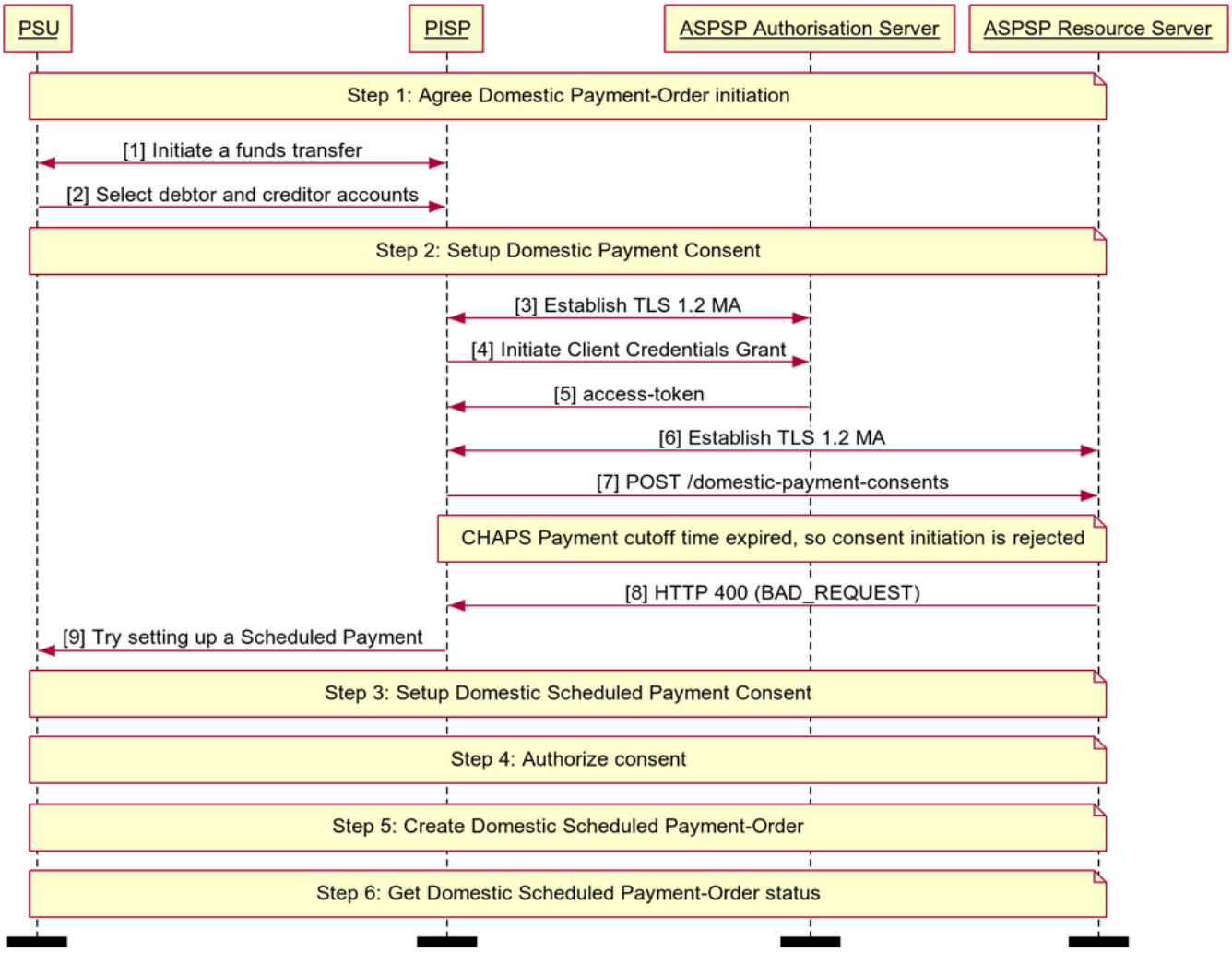
Multi-Auth Payment Order Consent


PSU Initial Authoriser
PSU Final Authoriser
PISP
ASPSP Authorisation Server
ASPSP Resource Server
Step 1: Agree domestic payment Initiation
[1] Agree domestic payment initiation request
Step 2: Setup Payment-Order Consent
[2] Establish TLS 1.2 MA
[3] Initiate Client Credential Grant
[4] access-token
[5] Establish TLS 1.2 MA
[6] POST /domestic-payment-consents
Consent Status: AwaitingAuthorisation
[7] HTTP 201 (Created),ConsentId
[8] HTTP 302 (Found),Redirect (ConsentId)
Step 3: Authorize Consent - Initial Authoriser
[9] Follow redirect (ConsentId)
[10] Authenticate (SCA if required) and Authorise consent
Consent Status: Authorised
[11] HTTP 302 (Found), Redirect (authorization-code)
[12] Follow redirect (authorization-code)
[13] Establish TLS 1.2 MA
[14] Exchange authorization-code for access token
[15] access-token
Step 4: Create Payment-Order
[16] Establish TLS 1.2 MA
[17] POST /domestic-payments
[18] HTTP 201 (Created) DomesticPaymentId
Consent Status: Consumed
Payment Status: Pending
MultiAuthorisation Status: AwaitingFurtherAuthorisation
Step 5: Authorize Consent - Final Authoriser
[19] Authenticate (SCA if required) and Authorise consent
MultiAuthorisation Status: Authorised
Payment Status: AcceptedSettlementInProcess
Payment Status: AcceptedSettlementComplete
alt
[If PISP has registered a URL for event notification]
[20] POST /event-notifications
[21] HTTP 200 (OK)
[22] GET /domestic-payments/{DomesticPaymentId}
[23] HTTP 200 (OK) domestic-payment resource
[PISP may poll payment order status]
loop
[24] GET /domestic-payments/{DomesticPaymentId}
[25] HTTP 200 (OK) domestic-payment resource

Multi-Auth Payment Order Consent - Sequence Diagram

Reject the Payment Order Consent Creation After CutOffDateTime

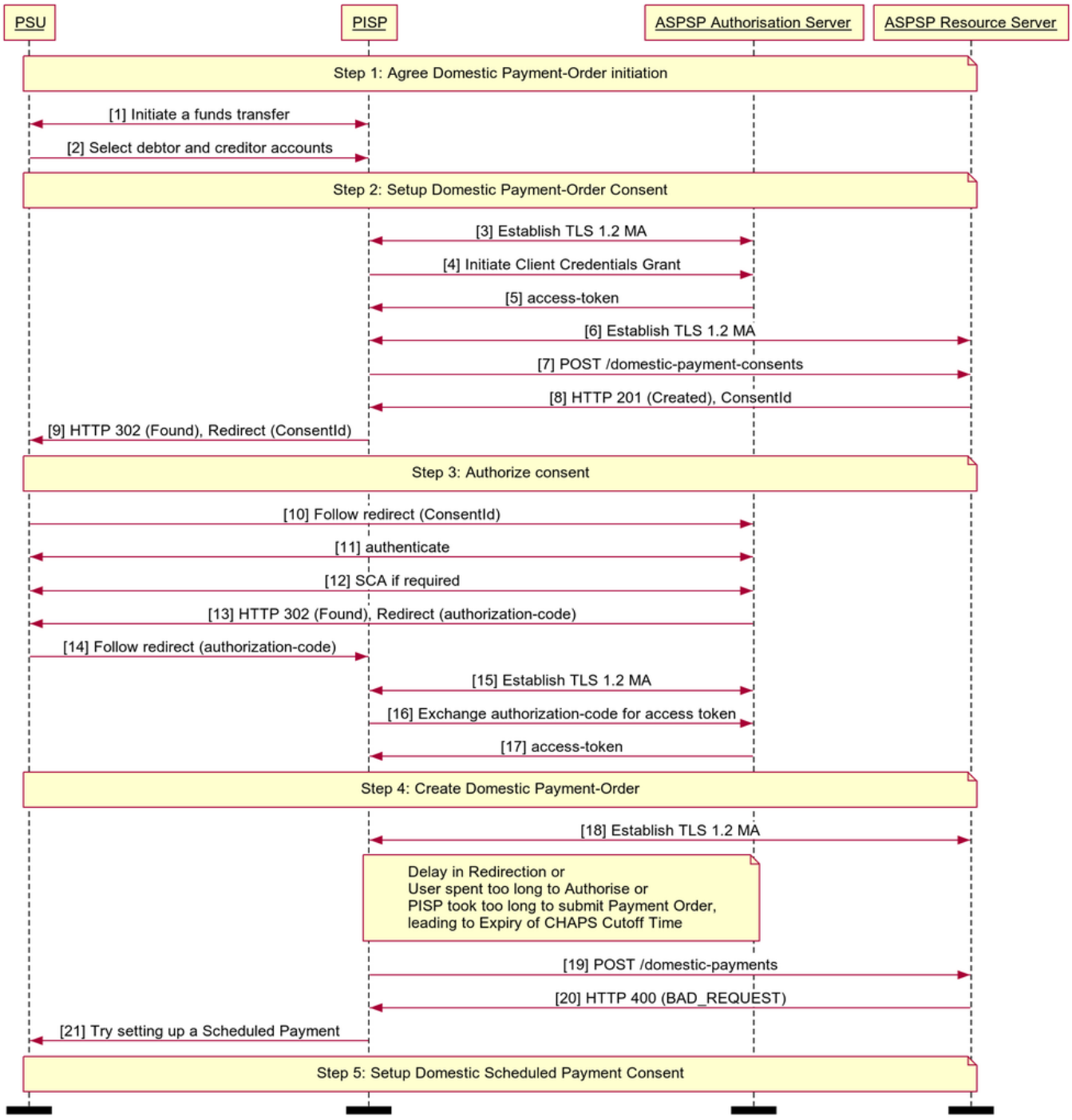
This example illustrates a scenario where an ASPSP choses to Reject the Payment-Order consent/resource request, after the CutoffTime. We have a CHAPS payment-order consent created after the CutOffDateTime, and ASPSP rejects the Consent, and the PISP chooses to place a Scheduled Payment-Order consent.



Payment Order Conflict - Sequence Diagram

Reject the Payment Order Creation After CutOffDateTime

This example illustrates a scenario where an ASPSP choses to Reject the Payment-Order consent/resource request, after the CutoffTime. We have a CHAPS payment-order Consent created and the Authorisation completed before the CutOffDateTime, but the Payment-Order submission happened after the CutOffDateTime, so the ASPSP has rejected it.



Payment Order Conflict - Sequence Diagram