In addition, the response message might optionally contain a _links section containing a hyperlink to tell the TPP the next step to avoid further errors, cp. Section 4.15. This applies especially in case of PSU authentication errors where a resubmission of credentials by the TPP might be needed after new entering of credentials by the PSU.

Response Code

The HTTP response code is 4xx or 5xx as defined in Section 4.12 for response codes in case of errors.

Response Header

Attribute	Type	Condition	Description
Content-Type	String	Mandatory	The string application/json is used.

Response Body

Attribute	Type	Condition	Description
tppMessages	Array of TPP Message Information	Optional	Error information
_links	Links	Optional	Should refer to next steps if the problem can be resolved e.g. for re-submission of credentials.

Example 1 (Access token not correct):

```
{ "tppMessages": [{
        "category": "ERROR",
        "code": "TOKEN_INVALID",
        "text": "additional text information of the ASPSP up to 500
characters"
   }]
}
```

Example 2 (Password incorrect):

```
{ "tppMessages": [{
        "category": "ERROR",
        "code": "PSU_CREDENTIALS_INVALID",
        "text": "additional text information of the ASPSP up to 500 
characters"
   }],
  "_links": {
        "updatePsuAuthentication": {"href": "/psd2/v1/payments/sepa-credit-
transfers/1234-wertiq-983/authorisations/123auth456"}
            }
}
```

4.13.3.2 Standardised Additional Error Information

In [RFC7807], a standardised definition of reporting error information is described. In the following, requirements of how to use this standardised error information reporting in the context of the NextGenPSD2 XS2A interface are defined.

Response Code

The HTTP response code is 4xx or 5xx as defined in Section 4.12 for response codes in case of errors. However, with the same reasoning as in Section 4.13.3.1, Additional Error Information may also be included in certain responses with positive response codes.

Response Header

Attribute	Type	Condition	Description
Content-Type	String	Mandatory	The string application/problem+json is used.

Response Body

Attribute	Type	Condition	Description
type	Max70Text	Mandatory	A URI reference [RFC3986] that identifies the problem type. **Remark for Future**: These URI will be provided by NextGenPSD2 in future.
title	Max70Text	Optional	Short human readable description of error type. Could be in local language. To be provided by ASPSPs.
detail	Max500Text	Optional	Detailed human readable text specific to this instance of the error. XPath might be

Attribute	Type	Condition	Description
			used to point to the issue generating the error in addition. **Remark for Future**: In future, a dedicated field might be introduced for the XPath.
code	Message Code	Mandatory	Message code to explain the nature of the underlying error.
additionalErrors	Array of Error Information	Optional	Might be used if more than one error is to be communicated
_links	Links	Optional	Should refer to next steps if the problem can be resolved e.g. for re-submission of credentials.

Example

```
  HTTP/1.1 401 Unauthorized
  Content-Type: application/problem+json
  Content-Language: en
   {
    "type": "https://berlingroup.com/error-codes/TOKEN_INVALID",
    "title": " The OAuth2 token is associated to the TPP but is not valid
for the addressed service/resource.",
   "detail": " additional text information of the ASPSP up to 500
characters ",
    "code": "TOKEN_INVALID",
     "additionalErrors": [ {
            "title": "The PSU-Corporate-ID cannot be matched by the
addressed ASPSP.",
            "detail": "additional text information of the ASPSP up to 500
characters",
            "code": "CORPORATE_ID_INVALID"
    },… ],
    "_links": { }
   }
```

4.14 Status Information

4.14.1 Status Information for PIS

The backend systems of ASPSPs are supporting for payments a transaction status, which is defined in the ISO20022 and is addressed in this specification as the data element "transactionStatus". ASPSPs will deliver this status within all response messages after a payment initiation resource has been established and if no error occurs.

The transaction status of a payment initiation is changing during the initiation process, depending on the results of sub-steps like format checks, SCA checks, PSU related profile checks, funds availability checks or depending on the start of backend clearing processes. At the end of a payment process, the transaction status in the ASPSPs backend is either “RJCT”, which stands for "Rejected", or “ACSC", which stands for "AcceptedSettlementCompleted” where complete is here referring to the debtor account. For instant payments, the additional transaction status "ACCC", which stands for "AcceptedSettlementCompleted" regarding the creditor account might be used in addition. Depending on the booking process of the ASPSP, the risk of the actual payment, the financial account status of the PSU account or the initiation date and time, the latter status might be reached after some period and after the payment initiation process as such has been finalised. These later transaction statuses do not need to be reflected in the XS2A interface which is only providing the status information immediately after the initiation of the payment.

A typical end status with in PIS process for a batch booking process is therefore

- "ACTC" which stands for “AcceptedTechnicalCorrect”, where the PSU authentication, syntactical and semantical (product) checks had been successful,
- "ACWC" which stands for "AcceptedWithChanges", where the PSU authentication, syntactical and semantical (product) checks had been successful and the ASPSP is informing the PISP that some changes have been applied to the payment initiation, e.g. on the requested execution date,
- "ACCP“, which stands for "AcceptedCustomerProfile”, where in addition the financial risk profile of the PSU including funds availability has been checked positively, or
- "ACFC“, which stands for "AcceptedFundsChecked”, where in addition to the customer profile the funds availability has been checked positively.

Realtime booking processes for batch payments might result for the time period of the payment initiation in

- "ACSP", which stands for “AcceptedSettlementInProcess”, where the settlement routine regarding the debtor account of the payment has already been initiated.

- "ACSC", which stands for “AcceptedSettlementCompleted”, indicating that the money has been booked already from the debtor account.

For instant payments, the final backend status "ACCC", which stands for "AcceptedSettlementCompleted" regarding the creditor account, will normally be reported at the end of the payment initiation process.

In bulk payment initiation, ASPSPs might choose either to process the bulk only partially and reject some of the contained payments. This results in

- "PART", which stands for "PartiallyAccepted", indicating that all mandated authorisations have been applied, but not all payment have been transformed due to other reasons.

Funds Availability

For ASPSP, which are not booking the money directly from the account, this specification provides the optional data element

"fundsAvailable": true/false

to be used together with the codes "ACTC", "ACWC" and "ACCP" in a GET Status Response Message early in the process chain to indicate that a funds check has been processed with the indicated result. This is the same data element as used in the confirmation of funds request and might be used by the ASPSP to inform the PISP about the funds availability, following requirements from [EBA-RTS].

Even if the funds check has been positive, the payment might be rejected later during the batch booking phase due to other bookings on the account. In case of no funds available, the payment might not be rejected yet due to the practice of the ASPSP in online channels, that it will wait for liquidity for a certain period.

Example 1: Batch booking bank, no profile checks but funds available positive

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

Example 2: Batch booking bank, profile check positive, no funds available, no rejection yet

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

The ASPSP might also use the status "PDNG", which stands for “Pending” to inform the TPP about the fact, that the next status of the payment has not been reached yet.

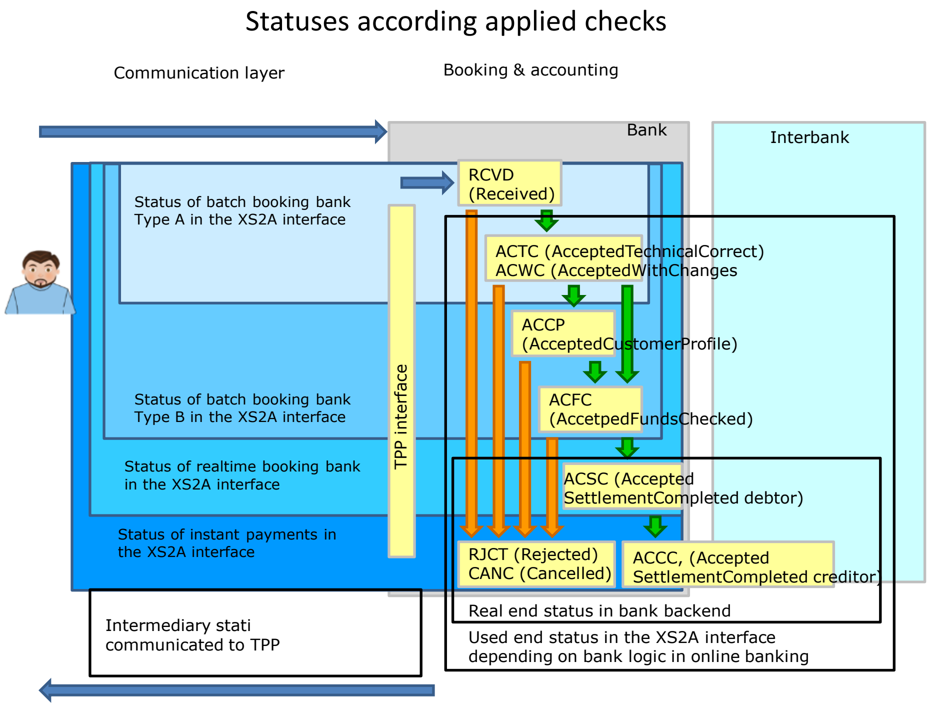
In addition, the ASPSP will inform the TPP about the status of the technical SCA process for a payment initiation within the GET SCA Status Response Message. For this status reporting the data element “scaStatus” is used.

Future dated payments and periodic payments

Future dated payments and periodic payments are both payment types which are not directly executed after initiation. For both types of payments, ASPSPs might have a reduced or no check on customer profile or funds availability due to the fact that the actual payments are performed later. The end status during the payment initiation process then is "ACTC" or "ACCP" depending on the ASPSPs procedures in its online channels. The fundsAvailable data element might be contained in addition, in case a funds availability check has been performed during payment initiation.

Status Model Overview

The following picture gives an overview on the transaction status:


Statuses according applied checks
Communication layer
Booking & accounting
Bank
Interbank
RCVD (Received)
Status of batch booking bank Type A in the XS2A interface
ACTC (AcceptedTechnicalCorrect)
ACWC (AcceptedWithChanges
ACCP (AcceptedCustomerProfile)
Status of batch booking bank Type B in the XS2A interface
ACFC (AcceptedFundsChecked)
TPP interface
Status of realtime booking bank in the XS2A interface
ACSC (Accepted SettlementCompleted debtor)
Status of instant payments in the XS2A interface
RJCT (Rejected)
CANC (Cancelled)
ACCC, (Accepted SettlementCompleted creditor)
Real end status in bank backend
Intermediary stati communicated to TPP
Used end status in the XS2A interface depending on bank logic in online banking

Status of cancelled Payments

After a successful cancellation of a payment initiation, the corresponding transaction status transforms to "CANC" for cancelled. This transaction status will be returned as long as the cancelled payment initiation resource is addressable.

Remark: This code is not yet part of the ISO20022 transaction status external reason code. The Berlin Group will raise a corresponding change request.

Status of partially authorised payments within a multilevel SCA process

Payment initiations which are at least authorised by one PSU, but which are not yet finally authorised by all applicable PSU will be transformed into the new status "PATC" for "PartiallyAcceptedTechnicalCorrect".

4.14.2 Status Information for the AIS within the Establish Consent Process

The status of the consent resource is changing during the initiation process as well as the transaction status of a payment initiation resource. In difference to the payment initiation process, there are only SCA checks on the consent resource and no feedback loop with the ASPSP backend. The data element for the status of the consent is defined as "consentStatus".

The only codes within the **initiation phase** supported for the consentStatus for this process are "received", “rejected”, "partiallyAuthorised”and “valid”. The current status of the consent resource is returned within all response messages during the authorisation process of the consent.

After a successful authorisation of a consent by a PSU, the consent resource might change its status during its lifecycle which needs to be transparent to the AIS. The following codes are supported during the **lifecycle phase** of the consent:

- "expired": The consent has been expired (e.g. after the period of time which is mandated by an applicable regulation or after the period time requested by the TPP).
- "revokedByPsu": The consent has been revoked by the PSU.
- "terminatedByTpp": The AIS has terminated the consent.

The AIS can retrieve this status within the GET Status Response Message.

Note: The "expired" status will also apply to one off consents, once they are used or out dated.

Note: The "terminatedByTpp" status will also apply, when a recurring has been terminated in case of a side effect by the same TPP establishing a new consent for the same PSU.

In addition, the ASPSP informs the TPP about the status of the technical SCA process for establishing a consent within the GET SCA Status Response Message. For this status reporting the data element “scaStatus” will be used.

4.15 API Steering Process by Hyperlinks

The XS2A API requires for the payment initiation and account information service several requests from the TPP towards the ASPSP. With the Payment Initiation Request and the Account Information Consent Request, a resource presentation is generated by the ASPSP. The location header of the response will usually contain a link to the created resource.

In addition, the ASPSP can embed a hyperlink together with a "tag" for the semantics of this hyperlink into the response to these first requests and to all succeeding requests within the services. This hyperlink must be a URI reference as defined in [RFC3986] and can be either a relative link, which is recommend to save space, for the host starting e.g. with "/psd2/v1/payments/sepa-credit-transfers" or it can be a global link like https://www.testbank.com/psd2/v1/payments/sepa-credit-transfers/asdf-asdf-asdf-1234.

The global links might be needed in some circumstances, e.g. a re-direct. The tag of the hyperlink transports the functionality of the resource addressed by the link, e.g. "authorise-transaction". This link indicates that results of a SCA method are to be posted to the resource addressed by this link to authorise e.g. a payment.

The steering hyperlinks are transported in the "_links" data element, cp. [HAL]. It may contain one or several hyperlinks.

The "_links" data element may contain more hyperlinks than specified in the related call. In this case, this will be documented in the ASPSP's PSD2 documentation or the hyperlinks can be ignored by the TPP.

In Section 14.6, the list of supported hyperlink types is defined.

Some hyperlinks might require additional data in the same response body which are then needed when following this hyperlink. The following table gives an overview on these specific steering hyperlinks to explain interconnection with the data elements.

Hyperlink	Additional Link Related Data	Description
startAuthorisationWith PsuAuthentication	(challengeData)	The link to an endpoint where the authorisation of a transaction or of a transaction cancellation shall be started, where PSU authentication data shall be uploaded with the corresponding call. Remark: In rare cases the ASPSP will ask only for some dedicated ciphers of the passwords. This information is then transported to the TPP by using the "challenge" data element, normally used only in SCA context.
startAuthorisationWith EncryptedPsuAuthentication	(challengeData)	Same as startAuthorisactionWith PsuAuthentication, but password is encrypted on application layer when uploaded.
updatePsuAuthentication	(challengeData)	The link to the payment initiation/consent resource, which needs to be updated by a PSU password and eventually the PSU identification if not delivered yet. **Remark**: In rare cases the ASPSP will ask only for some dedicated ciphers of the passwords. This information is then transported to the TPP by using the "challenge" data element, normally used only in SCA context.
updateEncryptedPsu Authentication	(challengeData)	Same as updatePsuAuthentication, but password is encrypted on application layer when uploaded.

Hyperlink	Additional Link Related Data	Description
startAuthorisationWith AuthenticationMethodSelection	scaMethods	This is a link to and endpoint where the authorisation of a transaction or of a transaction cancellation shall be started, where the selected SCA method shall be uploaded with the corresponding call.
selectAuthenticationMethod	scaMethods	This is a link to a resource, where the TPP can select the applicable strong customer authentication methods for the PSU, if there were several available authentication methods.
authoriseTransaction	challengeData, chosenScaMethod	A link to the resource, where a "Transaction Authorisation Request" can be sent to. This request transports the result of the SCA method performed by the customer, generating a response to the challenge data.
startAuthorisationWith TransactionAuthorisation	challengeData, chosenSCAMethod	A link to an endpoint, where an authorisation of a transaction or a cancellation can be started, and where the response data for the challenge is uploaded in the same call for the transaction authorisation or transaction cancellation at the same time in the Embedded SCA Approach.

4.16 Data Extensions

The ASPSP might add more data attributes to response messages. Such extensions then shall be documented in the ASPSP's documentation of its XS2A interface. These data attributes can be either ignored by the TPP or can be interpreted as defined by the above mentioned documentation.

The ASPSP might add additional optional data attributes to be submitted, e.g. for setting up additional services. In addition, an ASPSP can ask the TPP for a submission of proprietary data in a second step via the "proprietaryData" hyperlink. This shall be published by the ASPSP in its documentation.

> **Remark**: Before defining these additional proprietary data elements, the ASPSP is requested to submit the attribute description to the Berlin Group NextGen Taskforce, where it will be decided on a standardised approach for the related data attributes.

5 Payment Initiation Service

Remark: The API design differs across the various SCA approaches (Embedded, Redirect, OAuth2 or Decoupled, cp. [XS2A-OR]), but most between the Embedded SCA Approach and the others, since the Embedded SCA Approach demands the support of the full SCA complexity within the API itself. For that reason, all data or processes, which are needed for the Embedded SCA Approach only, are shown with a light blue background, to increase the readability of the specification.

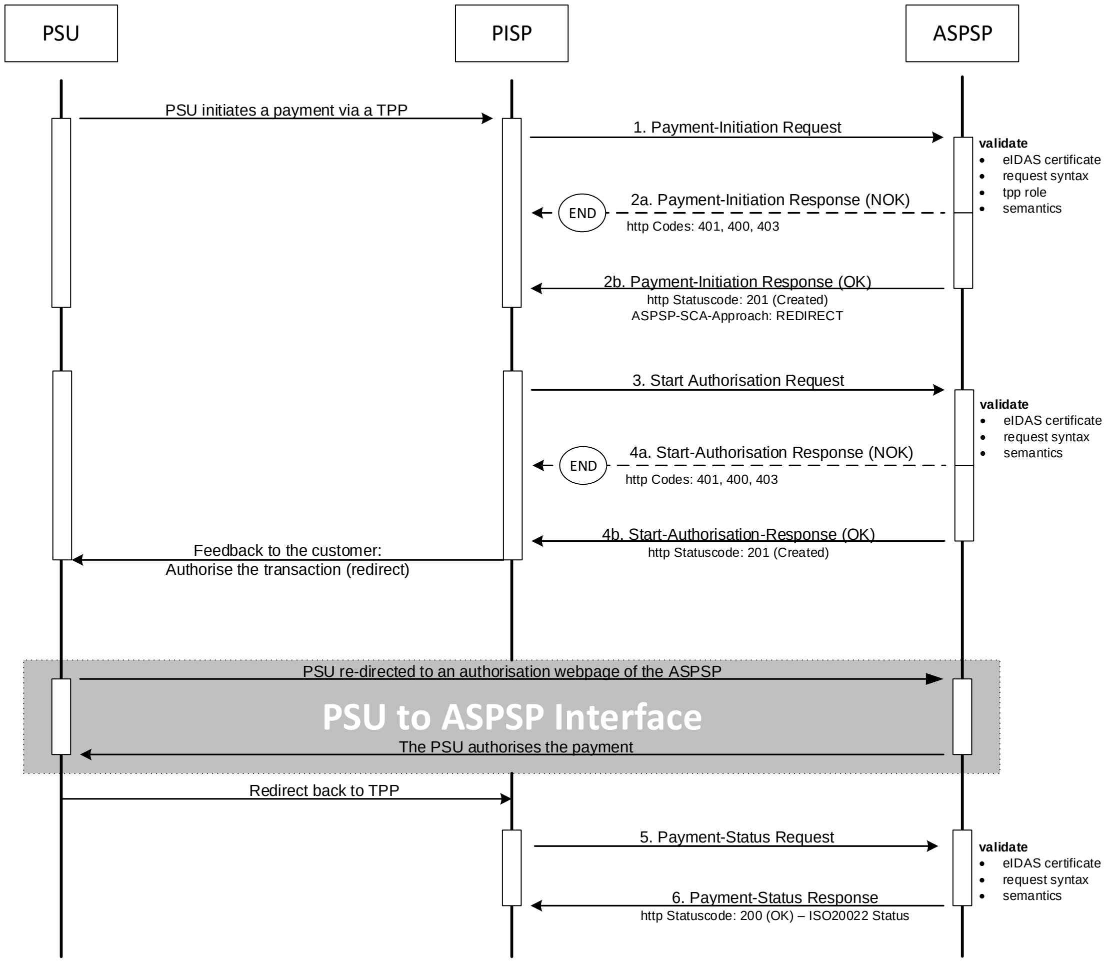
5.1 Payment Initiation Flows

The payment initiation flow depends heavily on the SCA approach implemented by the ASPSP. The most complex flow is the flow for the Embedded SCA Approach, which further differs on whether there are various authentication methods available for the PSU. In the following, the different API flows are provided as an overview for these different scenarios.

> **Remark:** The flows do not always cover all variances or complexities of the implementation and are exemplary flows.

5.1.1 Redirect SCA Approach: Explicit Start of the Authorisation Process

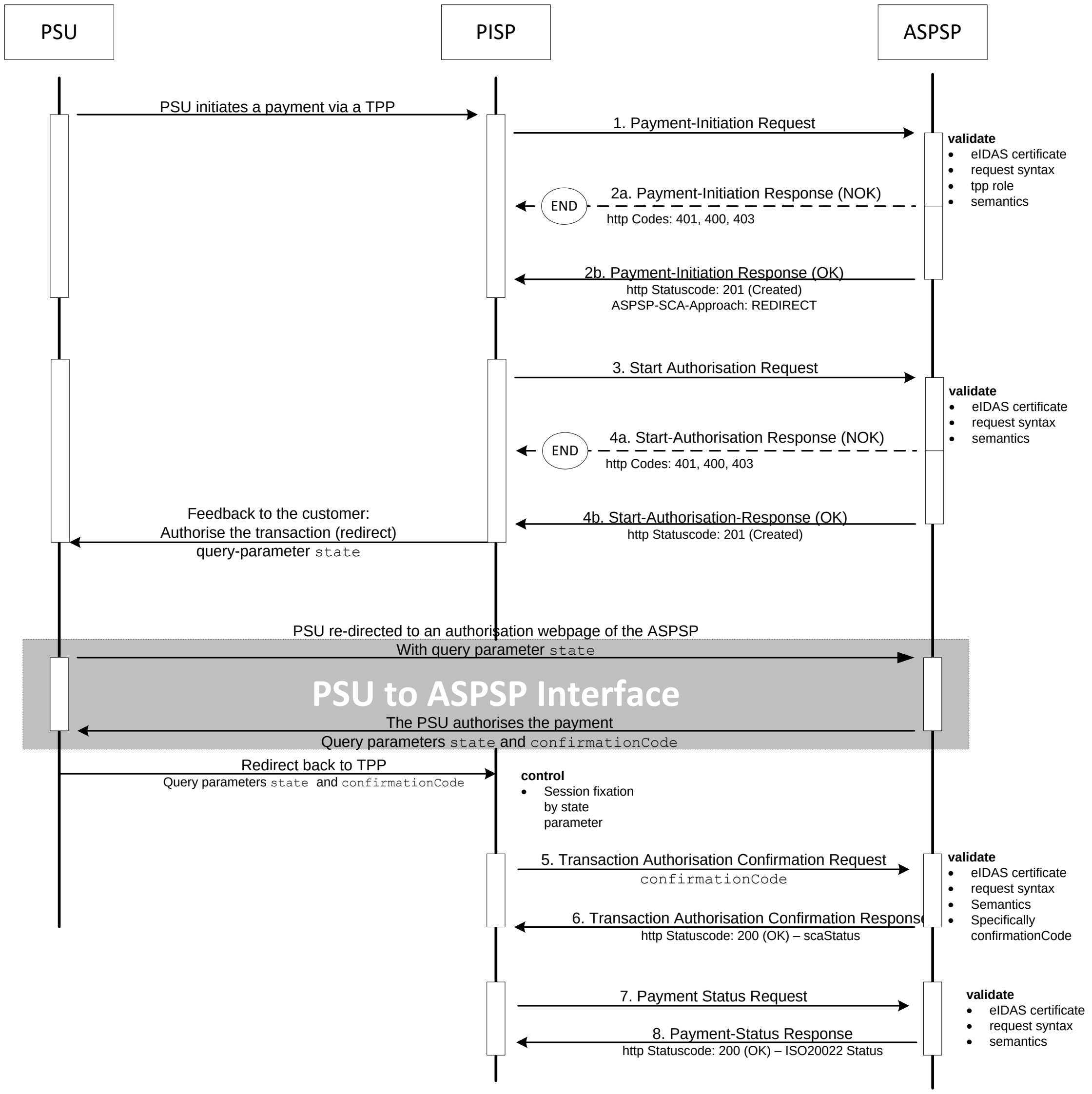
If the ASPSP supports the Redirect SCA Approach, the message flow within the payment initiation service is simple. The Payment Initiation Request is followed by an explicit request of the TPP to start the authorisation. This is followed by a redirection to the ASPSP SCA authorisation site. A status request might be requested by the TPP after the session is re-redirected to the TPP's system.


PSU
PISP
ASPSP
PSU initiates a payment via a TPP
1. Payment-Initiation Request
validate
• eIDAS certificate
• request syntax
• tpp role
• semantics
END
2a. Payment-Initiation Response (NOK)
http Codes: 401, 400, 403
2b. Payment-Initiation Response (OK)
http Statuscode: 201 (Created)
ASPSP-SCA-Approach: REDIRECT
3. Start Authorisation Request
validate
• eIDAS certificate
• request syntax
• semantics
END
4a. Start-Authorisation Response (NOK)
http Codes: 401, 400, 403
4b. Start-Authorisation-Response (OK)
http Statuscode: 201 (Created)
Feedback to the customer:
Authorise the transaction (redirect)
PSU re-directed to an authorisation webpage of the ASPSP
PSU to ASPSP Interface
The PSU authorises the payment
Redirect back to TPP
5. Payment-Status Request
validate
• eIDAS certificate
• request syntax
• semantics
6. Payment-Status Response
http Statuscode: 200 (OK) – ISO20022 Status

5.1.2 Redirect SCA Approach: Explicit Start of the Authorisation Process with Confirmation Code

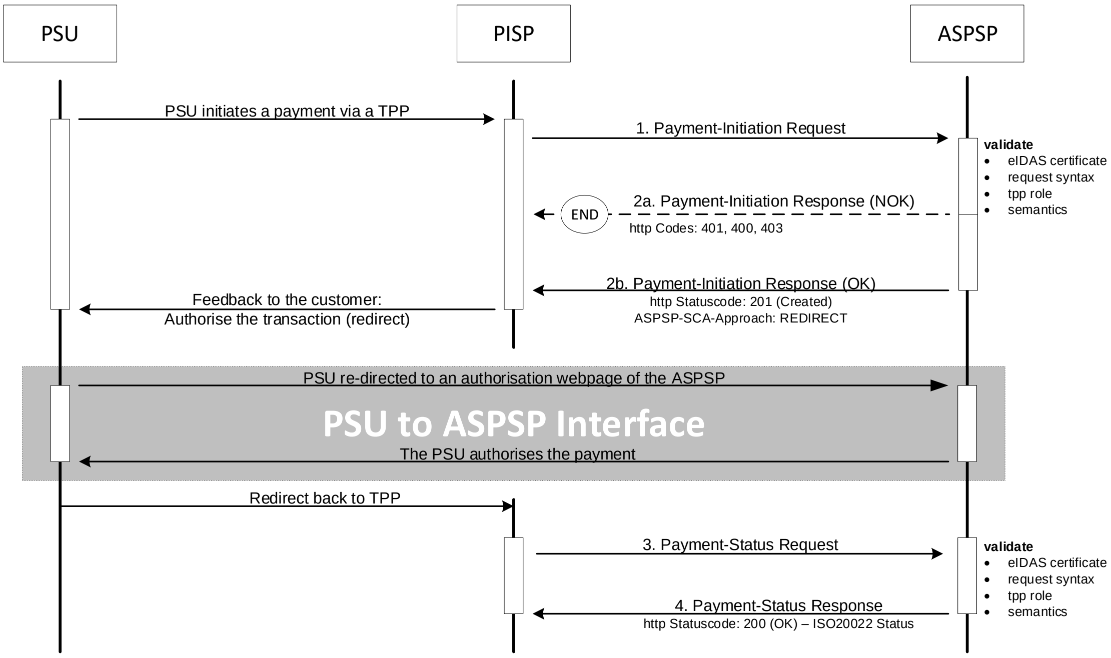
In addition to the scenario above, an authorisation confirmation request might be requested by the ASPSP from the TPP after the session is re-redirected to the TPP's system and after the

TPP's control on session fixation. In the end, a payment status request might be needed by the TPP to control the exact status of the payment initiation.


PSU
PISP
ASPSP
PSU initiates a payment via a TPP
1. Payment-Initiation Request
validate
• eIDAS certificate
• request syntax
• tpp role
• semantics
END
2a. Payment-Initiation Response (NOK)
http Codes: 401, 400, 403
2b. Payment-Initiation Response (OK)
http Statuscode: 201 (Created)
ASPSP-SCA-Approach: REDIRECT
3. Start Authorisation Request
validate
• eIDAS certificate
• request syntax
• semantics
END
4a. Start-Authorisation Response (NOK)
http Codes: 401, 400, 403
Feedback to the customer:
Authorise the transaction (redirect)
query-parameter state
4b. Start-Authorisation-Response (OK)
http Statuscode: 201 (Created)
PSU re-directed to an authorisation webpage of the ASPSP
With query parameter state
PSU to ASPSP Interface
The PSU authorises the payment
Query parameters state and confirmationCode
Redirect back to TPP
Query parameters state and confirmationCode
control
• Session fixation by state parameter
5. Transaction Authorisation Confirmation Request
confirmationCode
validate
• eIDAS certificate
• request syntax
• Semantics
• Specifically confirmationCode
6. Transaction Authorisation Confirmation Response
http Statuscode: 200 (OK) – scaStatus
7. Payment Status Request
validate
• eIDAS certificate
• request syntax
• semantics
8. Payment-Status Response
http Statuscode: 200 (OK) – ISO20022 Status

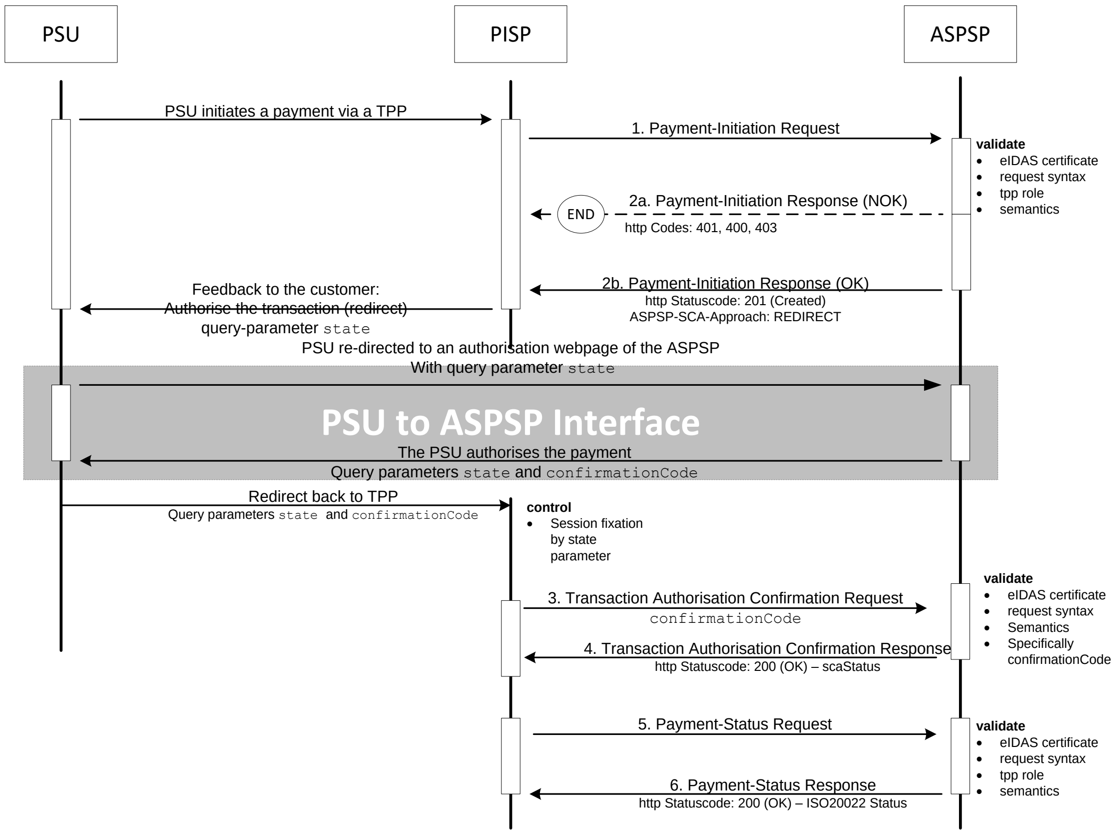
5.1.3 Redirect SCA Approach: Implicit Start of the Authorisation Process

ASPSPs might start the authorisation process implicitly in case of no additional data is needed from the TPP. This optimisation process results in the following flow (which is exactly the Redirect SCA Approach flow from the version 1.0 and 1.1 of the Implementation Guideline before authorisation sub-resources have been established). In this case, the redirection of the PSU browser session happens directly after the Payment Initiation Response. In addition an SCA status request can be sent by the TPP to follow the SCA process (not shown in the diagram).


PSU
PISP
ASPSP
PSU initiates a payment via a TPP
1. Payment-Initiation Request
validate
eIDAS certificate
request syntax
tpp role
semantics
END
2a. Payment-Initiation Response (NOK)
http Codes: 401, 400, 403
2b. Payment-Initiation Response (OK)
http Statuscode: 201 (Created)
ASPSP-SCA-Approach: REDIRECT
Feedback to the customer:
Authorise the transaction (redirect)
PSU re-directed to an authorisation webpage of the ASPSP
PSU to ASPSP Interface
The PSU authorises the payment
Redirect back to TPP
3. Payment-Status Request
validate
eIDAS certificate
request syntax
tpp role
semantics
4. Payment-Status Response
http Statuscode: 200 (OK) – ISO20022 Status

5.1.4 Redirect SCA Approach: Implicit Start of the Authorisation Process with Confirmation Code

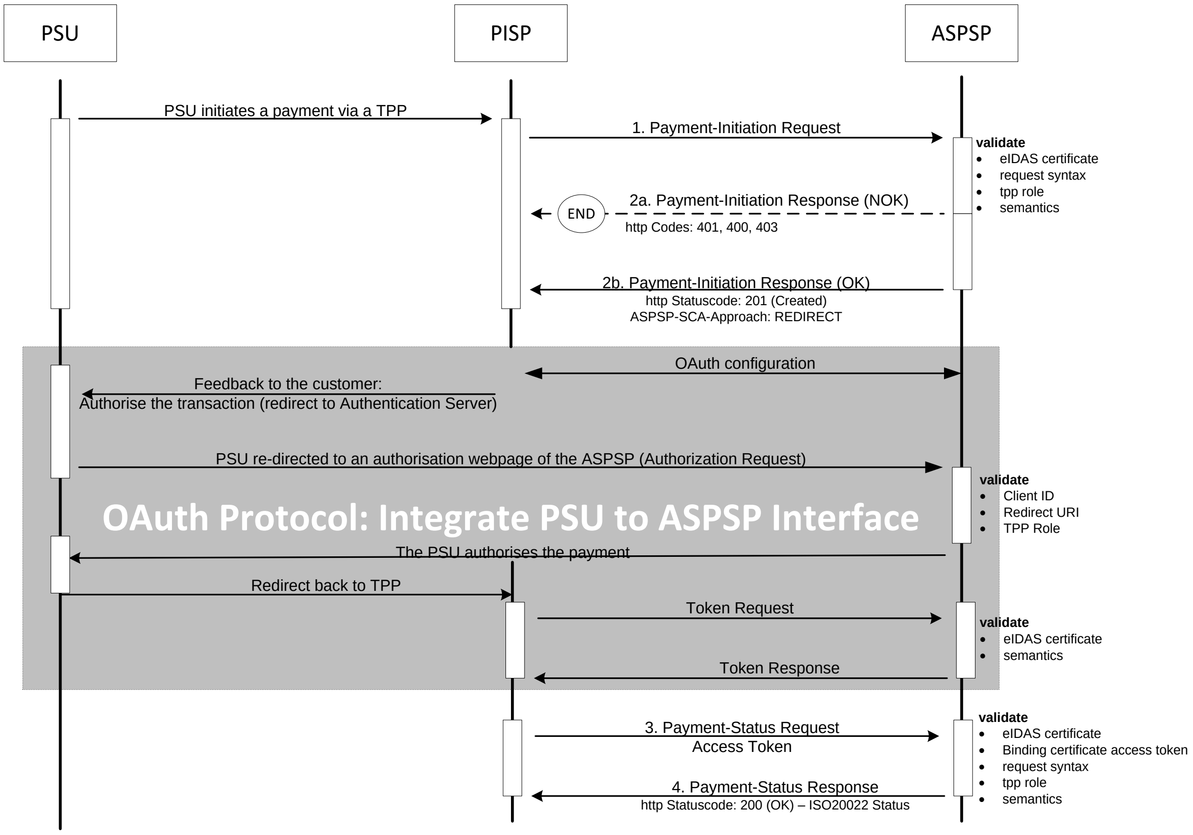
In addition to the scenario above, an authorisation confirmation request might be requested by the ASPSP from the TPP after the session is re-redirected to the TPP’s system and after the TPP's control on session fixation. In the end, a payment status request might be needed by the TPP to control the exact status of the payment initiation.


PSU
PISP
ASPSP
PSU initiates a payment via a TPP
1. Payment-Initiation Request
validate
• eIDAS certificate
• request syntax
• tpp role
• semantics
2a. Payment-Initiation Response (NOK)
END
http Codes: 401, 400, 403
2b. Payment-Initiation Response (OK)
http Statuscode: 201 (Created)
ASPSP-SCA-Approach: REDIRECT
Feedback to the customer:
Authorise the transaction (redirect)
query-parameter state
PSU re-directed to an authorisation webpage of the ASPSP
With query parameter state
PSU to ASPSP Interface
The PSU authorises the payment
Query parameters state and confirmationCode
Redirect back to TPP
Query parameters state and confirmationCode
control
• Session fixation by state parameter
validate
• eIDAS certificate
• request syntax
• Semantics
• Specifically confirmationCode
3. Transaction Authorisation Confirmation Request
confirmationCode
4. Transaction Authorisation Confirmation Response
http Statuscode: 200 (OK) – scaStatus
5. Payment-Status Request
validate
• eIDAS certificate
• request syntax
• tpp role
• semantics
6. Payment-Status Response
http Statuscode: 200 (OK) – ISO20022 Status

5.1.5 OAuth2 SCA Approach: Implicit Start of the Authorisation Process

If the ASPSP supports the OAuth2 SCA Approach, the flow is very similar to the Redirect SCA Approach with implicit start of the Authorisation Process. Instead of redirecting the PSU directly to an authentication server, the OAuth2 protocol is used for the transaction authorisation process.

Remark: The OAuth2 SCA Approach with explicit start of the Authorisation Process is treated analogously.

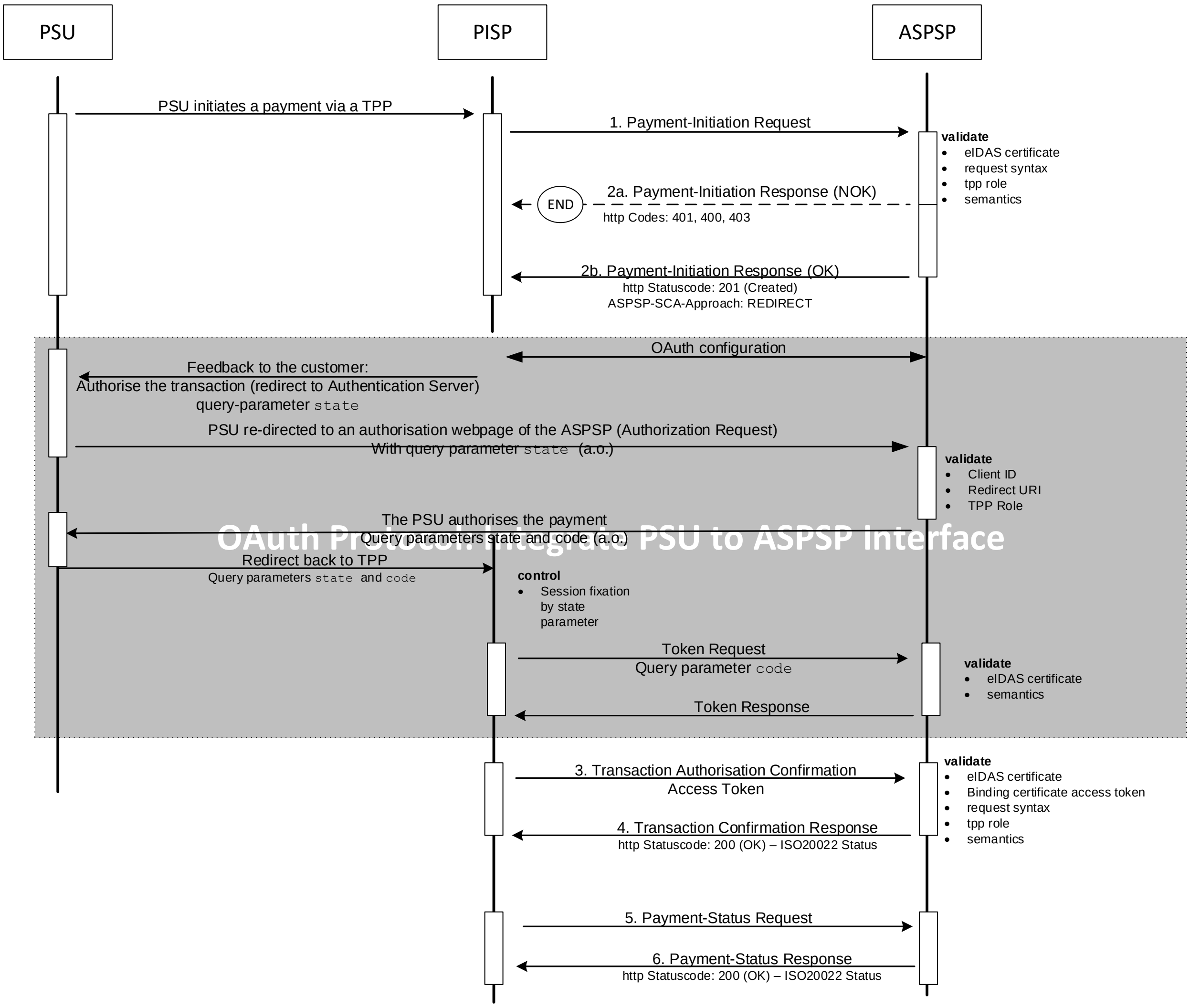

PSU
PISP
ASPSP
PSU initiates a payment via a TPP
1. Payment-Initiation Request
validate
eIDAS certificate
request syntax
tpp role
semantics
END
2a. Payment-Initiation Response (NOK)
http Codes: 401, 400, 403
2b. Payment-Initiation Response (OK)
http Statuscode: 201 (Created)
ASPSP-SCA-Approach: REDIRECT
OAuth configuration
Feedback to the customer:
Authorise the transaction (redirect to Authentication Server)
PSU re-directed to an authorisation webpage of the ASPSP (Authorization Request)
validate
Client ID
Redirect URI
TPP Role
OAuth Protocol: Integrate PSU to ASPSP Interface
The PSU authorises the payment
Redirect back to TPP
Token Request
validate
eIDAS certificate
semantics
Token Response
3. Payment-Status Request
Access Token
validate
eIDAS certificate
Binding certificate access token
request syntax
tpp role
semantics
4. Payment-Status Response
http Statuscode: 200 (OK) – ISO20022 Status

5.1.6 OAuth2 SCA Approach: Implicit Start of the Authorisation Process with Confirmation Code

In addition to the scenario above, an authorisation confirmation request might be requested by the ASPSP from the TPP after the session is re-redirected to the TPP's system and after the

TPP's control on session fixation. In the end, a payment status request might be needed by the TPP to control the exact status of the payment initiation.

Remark: The OAuth2 SCA Approach with explicit start of the Authorisation Process and with transaction confirmation step is treated analogously.

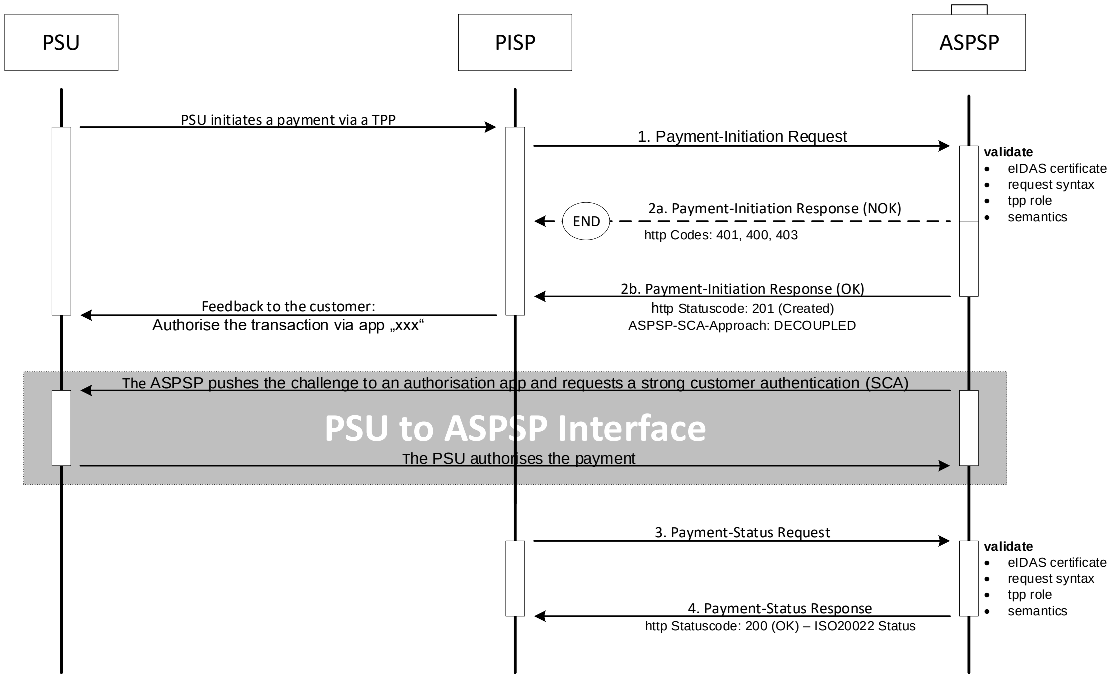

PSU
PISP
ASPSP
PSU initiates a payment via a TPP
1. Payment-Initiation Request
validate
• eIDAS certificate
• request syntax
• tpp role
• semantics
END
2a. Payment-Initiation Response (NOK)
http Codes: 401, 400, 403
2b. Payment-Initiation Response (OK)
http Statuscode: 201 (Created)
ASPSP-SCA-Approach: REDIRECT
OAuth configuration
Feedback to the customer:
Authorise the transaction (redirect to Authentication Server)
query-parameter state
PSU re-directed to an authorisation webpage of the ASPSP (Authorization Request)
With query parameter state (a.o.)
validate
• Client ID
• Redirect URI
• TPP Role
The PSU authorises the payment
Query parameters state and code (a.o.)
OAuth Protocol: Integrate PSU to ASPSP interface
Redirect back to TPP
Query parameters state and code
control
• Session fixation by state parameter
Token Request
Query parameter code
validate
• eIDAS certificate
• semantics
Token Response
3. Transaction Authorisation Confirmation
Access Token
validate
• eIDAS certificate
• Binding certificate access token
• request syntax
• tpp role
• semantics
4. Transaction Confirmation Response
http Statuscode: 200 (OK) – ISO20022 Status
5. Payment-Status Request
6. Payment-Status Response
http Statuscode: 200 (OK) – ISO20022 Status

It is further recommended for ASPSPs and TPPs in this case to follow the Security Best Practice definitions as defined in [OA-SecTop]. This reference will also be added in the next version of the Implementation Guidelines.

5.1.7 Decoupled SCA Approach: Implicit Start of the Authorisation Process

The transaction flow in the Decoupled SCA Approach is similar to the Redirect SCA Approach. The difference is that the ASPSP is asking the PSU to authorise the payment e.g. via a dedicated mobile app, or any other application or device which is independent from the online banking frontend. The ASPSP is asking the TPP to inform the PSU about this authentication by sending a corresponding PSU Message like "Please use your xxx App to authorise the payment".

After the SCA having been processed between ASPSP and PSU, the TPP then needs to ask for the result of the transaction. In the following, a flow with an implicit start of the authorisation process is shown:

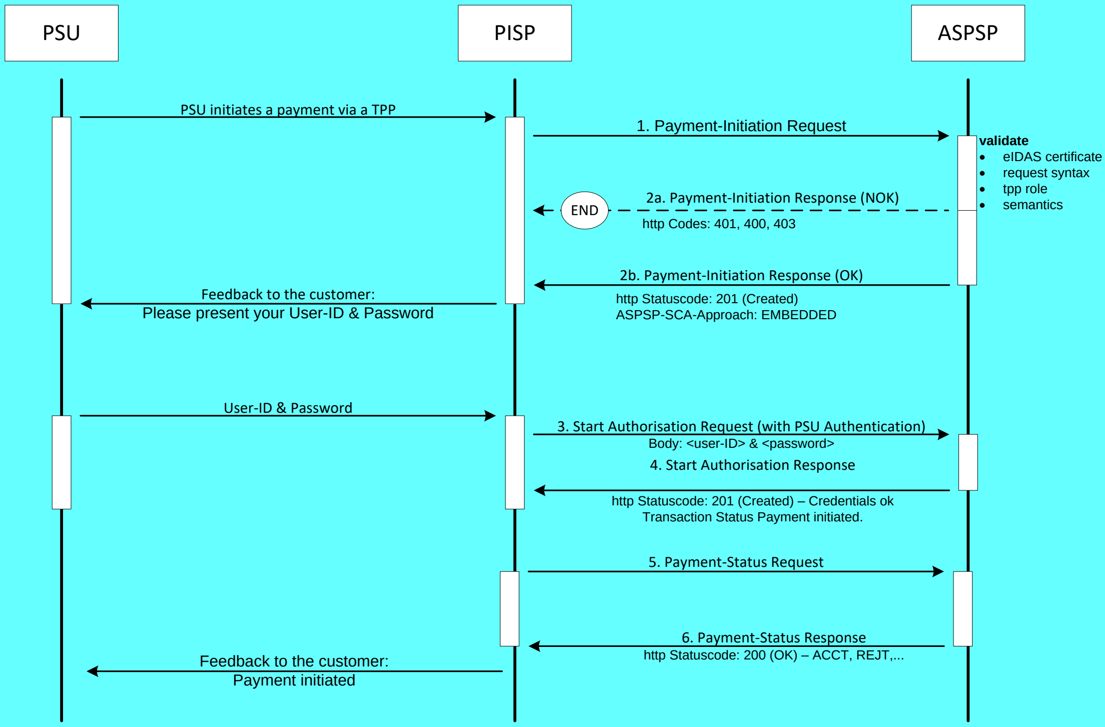

PSU
PISP
ASPSP
PSU initiates a payment via a TPP
1. Payment-Initiation Request
validate
eIDAS certificate
request syntax
tpp role
semantics
END
2a. Payment-Initiation Response (NOK)
http Codes: 401, 400, 403
2b. Payment-Initiation Response (OK)
http Statuscode: 201 (Created)
ASPSP-SCA-Approach: DECOUPLED
Feedback to the customer:
Authorise the transaction via app „xxx“
The ASPSP pushes the challenge to an authorisation app and requests a strong customer authentication (SCA)
PSU to ASPSP Interface
The PSU authorises the payment
3. Payment-Status Request
validate
eIDAS certificate
request syntax
tpp role
semantics
4. Payment-Status Response
http Statuscode: 200 (OK) – ISO20022 Status

Remark: In Section 6.1.1.3, a version with the explicit start of the Authorisation Process is documented for the Establish Consent Request.

5.1.8 Embedded SCA Approach without SCA method (e.g. Creditor in Exemption List)

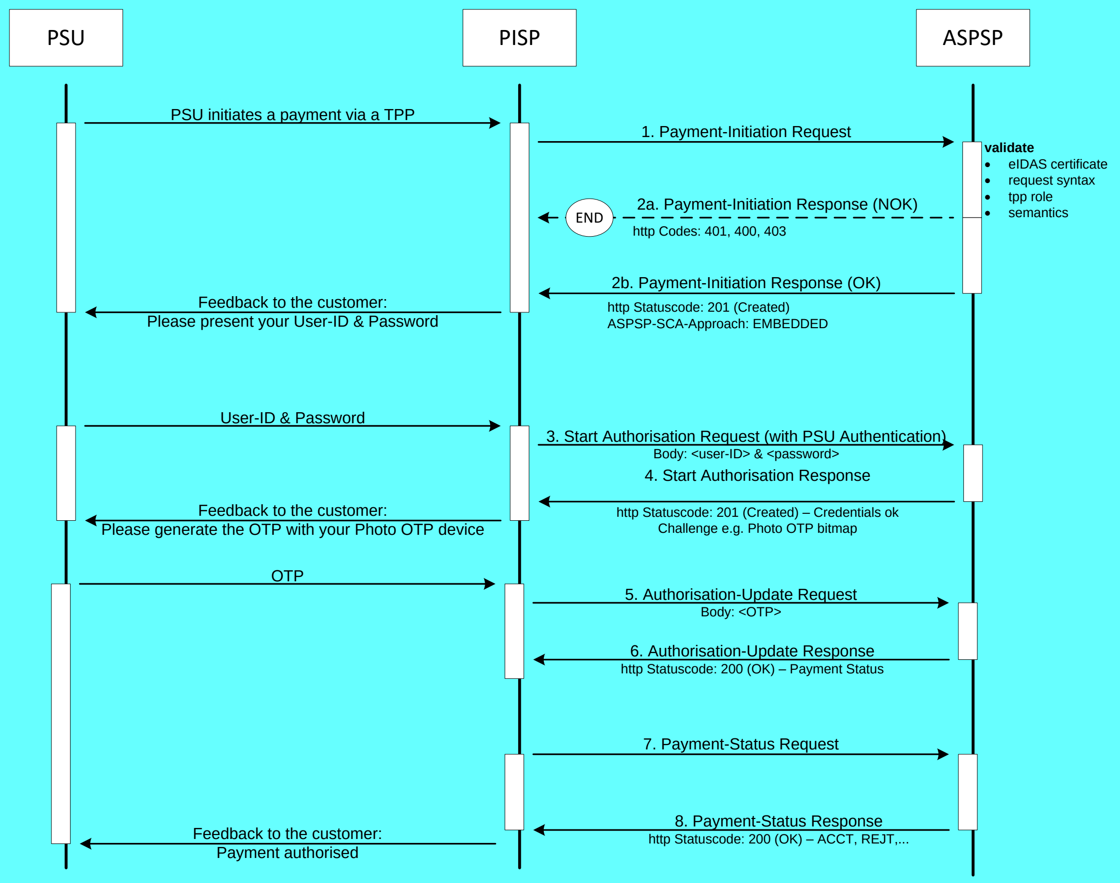
In the following, several exemplary flows are shown, where the ASPSP has chosen to process the SCA methods through the PISP – ASPSP interface. In any case, the PSU normally will need to authenticate himself with a first factor, before any account or SCA method details will be available to the PISP. So even in case where the Payment Initiation is accepted without an SCA method due e.g. to an exemption list, the PSU is asked via the PISP to provide the PSU Identification and e.g. a password or an OTP. The later exemplary flows then will show scenarios, where complexities like SCA processing and choosing an SCA method will be added.

Remark: In case where OAuth2 is requested by the ASPSP as a pre-step for PSU authentication, the sequence of the PSU authentication with the first authentication factor is omitted. This applies also for all examples for the Embedded SCA Approach.


PSU
PISP
ASPSP
PSU initiates a payment via a TPP
1. Payment-Initiation Request
validate
eIDAS certificate
request syntax
tpp role
semantics
END
2a. Payment-Initiation Response (NOK)
http Codes: 401, 400, 403
2b. Payment-Initiation Response (OK)
http Statuscode: 201 (Created)
ASPSP-SCA-Approach: EMBEDDED
Feedback to the customer:
Please present your User-ID & Password
User-ID & Password
3. Start Authorisation Request (with PSU Authentication)
Body: <user-ID> & <password>
4. Start Authorisation Response
http Statuscode: 201 (Created) – Credentials ok
Transaction Status Payment initiated.
5. Payment-Status Request
6. Payment-Status Response
http Statuscode: 200 (OK) – ACCT, REJT,...
Feedback to the customer:
Payment initiated

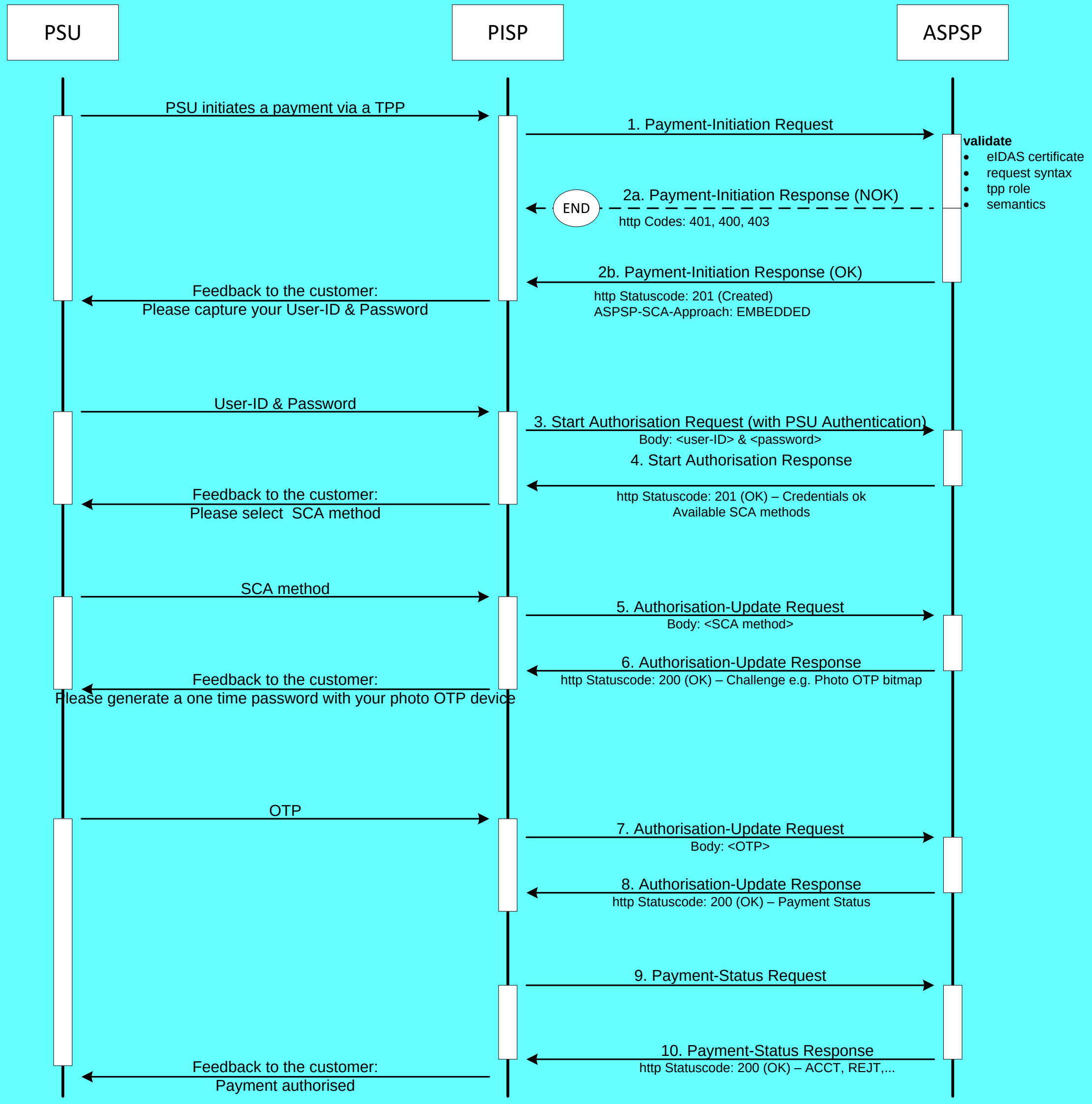
5.1.9 Embedded SCA Approach with only one SCA method available

In case where only one SCA method is available, the "Authorise Transaction Request" is added to the flow, where the TPP is transmitting the authentication data of the customer, e.g. an OTP with included dynamic linking to the transaction details.


PSU
PISP
ASPSP
PSU initiates a payment via a TPP
1. Payment-Initiation Request
validate
• eIDAS certificate
• request syntax
• tpp role
• semantics
END
2a. Payment-Initiation Response (NOK)
http Codes: 401, 400, 403
2b. Payment-Initiation Response (OK)
http Statuscode: 201 (Created)
ASPSP-SCA-Approach: EMBEDDED
Feedback to the customer:
Please present your User-ID & Password
User-ID & Password
3. Start Authorisation Request (with PSU Authentication)
Body: <user-ID> & <password>
4. Start Authorisation Response
http Statuscode: 201 (Created) – Credentials ok
Challenge e.g. Photo OTP bitmap
Feedback to the customer:
Please generate the OTP with your Photo OTP device
OTP
5. Authorisation-Update Request
Body: <OTP>
6. Authorisation-Update Response
http Statuscode: 200 (OK) – Payment Status
7. Payment-Status Request
8. Payment-Status Response
http Statuscode: 200 (OK) – ACCT, REJT,...
Feedback to the customer:
Payment authorised

5.1.10 Embedded SCA Approach with Selection of an SCA method

In the following flow, there is a selection of an SCA method added in case of the ASPSP supporting several SCA methods for the corresponding PSU. The ASPSP transmits first the available methods to the PISP. The PISP might filter them, if not all authentication methods can be technically supported. The available methods then are presented to the PSU for choice.


PSU
PISP
ASPSP
PSU initiates a payment via a TPP
1. Payment-Initiation Request
validate
eIDAS certificate
request syntax
tpp role
semantics
END
2a. Payment-Initiation Response (NOK)
http Codes: 401, 400, 403
2b. Payment-Initiation Response (OK)
http Statuscode: 201 (Created)
ASPSP-SCA-Approach: EMBEDDED
Feedback to the customer:
Please capture your User-ID & Password
User-ID & Password
3. Start Authorisation Request (with PSU Authentication)
Body: <user-ID> & <password>
4. Start Authorisation Response
http Statuscode: 201 (OK) – Credentials ok
Available SCA methods
Feedback to the customer:
Please select SCA method
SCA method
5. Authorisation-Update Request
Body: <SCA method>
6. Authorisation-Update Response
http Statuscode: 200 (OK) – Challenge e.g. Photo OTP bitmap
Feedback to the customer:
Please generate a one time password with your photo OTP device
OTP
7. Authorisation-Update Request
Body: <OTP>
8. Authorisation-Update Response
http Statuscode: 200 (OK) – Payment Status
9. Payment-Status Request
10. Payment-Status Response
http Statuscode: 200 (OK) – ACCT, REJT,...
Feedback to the customer:
Payment authorised

5.1.11 Combination of Flows due to mixed SCA Approaches

If an ASPSP supports for a PSU at least one decoupled SCA method and at the same time at least one SCA method that is not decoupled, then the above flows might be mixed as follows, since the ASPSP then needs to start the process with the assumption of one specific SCA approach to offer all available SCA methods to the PSU.

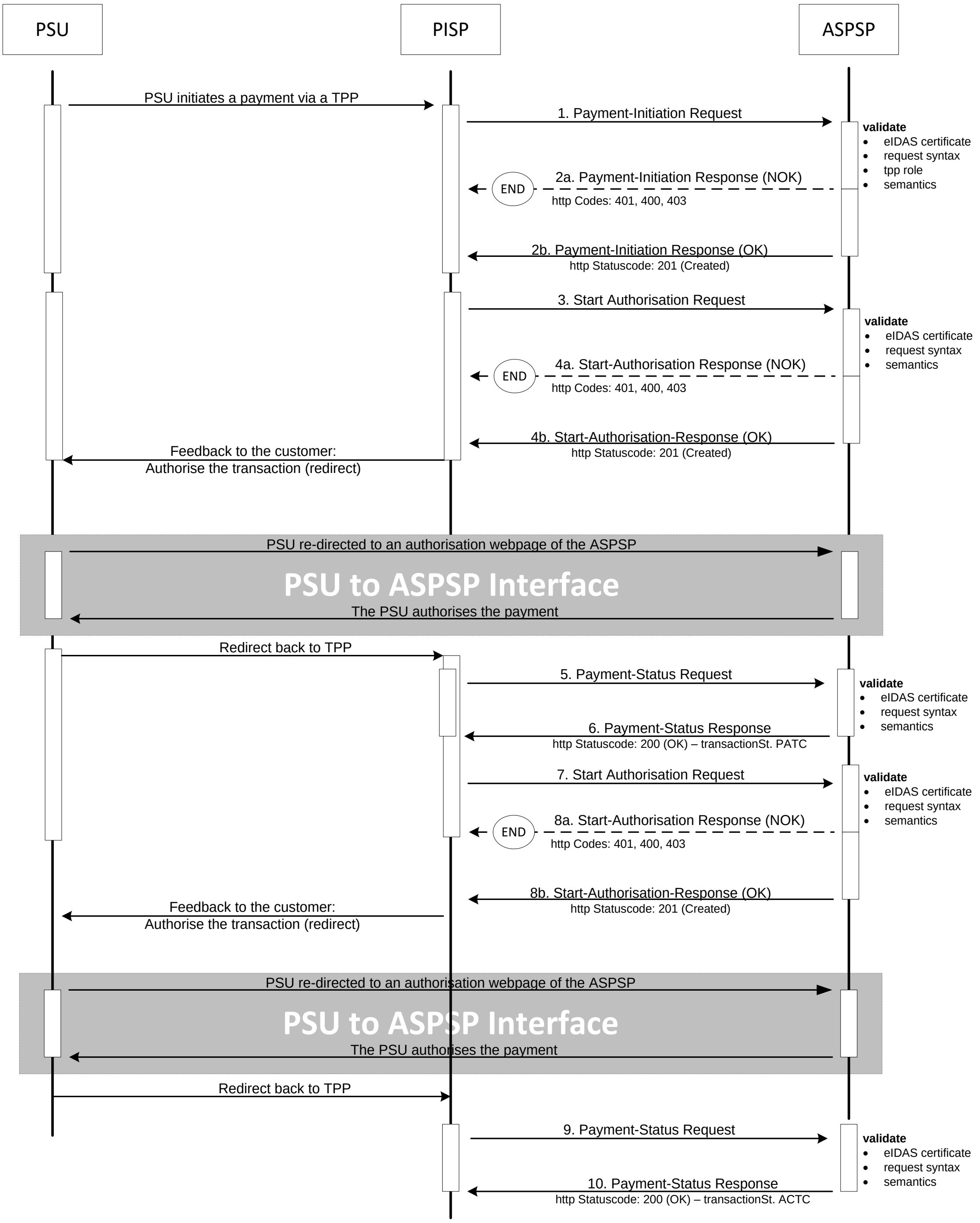
In case the ASPSP is starting the payment initiation flow with a redirect the PSU can choose on the authentication site of the ASPSP the decoupled authentication method. This is then transparent for the TPP and has no influence on the flows defined above.

In case the ASPSP is starting the payment initiation flow with the Embedded SCA Approach the ASPSP will provide a list of available SCA methods to the PSU via the TPP. If the PSU chooses an authentication method which requires the Decoupled SCA Approach, then the ASPSP is branching into the transaction flow for the Decoupled Approach as shown above: The ASPSP will return the corresponding HTTP header ASPSP-SCA-Approach with value "DECOUPLED" and the current status of the payment initiation, e.g. "ACTC" for correct technical checks but will return no hyperlink for further action other than the "self" and "status" hyperlink. The next request of the TPP then needs to be the GET Status Request to get the final status of the transaction after having processed the SCA method.

In case the ASPSP needs to decide between the Decoupled and the Redirect SCA approach, the ASPSP also might first offer the SCA methods available to the PSU and then branch after the selection of the PSU into the Decoupled or Redirect SCA Approach.

5.1.12 Multilevel SCA Approach: Example for the Redirect SCA Approach

The multilevel SCA Approach supports the authorisation of a payment by several users, e.g. in a 4 eyes principle authorisation. Multilevel SCA are always handled with Explicit start of the several Authorisation Mechanisms. In the following the flow for a 4 eyes principle authorisation is shown, where both SCA are performed by redirect.


PSU
PISP
ASPSP
PSU initiates a payment via a TPP
1. Payment-Initiation Request
validate
eIDAS certificate
request syntax
tpp role
semantics
END
2a. Payment-Initiation Response (NOK)
http Codes: 401, 400, 403
2b. Payment-Initiation Response (OK)
http Statuscode: 201 (Created)
3. Start Authorisation Request
validate
eIDAS certificate
request syntax
semantics
END
4a. Start-Authorisation Response (NOK)
http Codes: 401, 400, 403
4b. Start-Authorisation-Response (OK)
http Statuscode: 201 (Created)
Feedback to the customer:
Authorise the transaction (redirect)
PSU re-directed to an authorisation webpage of the ASPSP
PSU to ASPSP Interface
The PSU authorises the payment
Redirect back to TPP
5. Payment-Status Request
validate
eIDAS certificate
request syntax
semantics
6. Payment-Status Response
http Statuscode: 200 (OK) – transactionSt. PATC
7. Start Authorisation Request
validate
eIDAS certificate
request syntax
semantics
END
8a. Start-Authorisation Response (NOK)
http Codes: 401, 400, 403
8b. Start-Authorisation-Response (OK)
http Statuscode: 201 (Created)
Feedback to the customer:
Authorise the transaction (redirect)
PSU re-directed to an authorisation webpage of the ASPSP
PSU to ASPSP Interface
The PSU authorises the payment
Redirect back to TPP
9. Payment-Status Request
validate
eIDAS certificate
request syntax
semantics
10. Payment-Status Response
http Statuscode: 200 (OK) – transactionSt. ACTC

Remark: This flow is not depending on the SCA Approach. Multilevel SCA transactions are performed by using n times the Start Authorisation Request for n times SCA, where the corresponding SCA flow is replacing the Redirect SCA flow above. These SCA processes could also be performed simultaneously.

5.2 Data Overview Payment Initiation Service

The following table defines the technical description of the abstract data model as defined in [XS2A-OR] for the Payment Initiation service. The columns give an overview on the API protocols as follows:

- The "Data element" column is using the abstract data elements following [XS2A-OR] to deliver the connection to rules and role definitions in this document.
- The "Attribute encoding" is giving the actual encoding definition within the XS2A API as defined in this document.
- The "Location" columns define, where the corresponding data elements are transported as HTTP parameters on path, header or body level, resp. are taken from eIDAS certificates.

  Remark: Please note that website authentication certificate related data elements are not elements of the actual API call. They are indicated here, since they are mandated in the backend processing and might be transported from the API endpoint internally to the backend on the application layer. Please note, that in difference to this, the certificate data for the electronic seal can be transported within a dedicated HTTP header field.
- The "Usage" column gives an overview on the usage of data elements in the different services and API Calls. Within [XS2A-OR], the XS2A calls are described as abstract API calls. These calls will be technically realised as HTTPS POST, PUT and GET commands. The calls are divided into the following calls for Payment Initiation:
  - The Initiation Request which shall be the first API Call for every transaction within the corresponding XS2A service Payment Initiation. This call generates the corresponding resource within the Payment Initiation Service. The Payment Initiation can address a single payment, bulk payments and recurring payments. The latter are implemented as an initiation of a standing order.
  - The Update Data Call is a call, where the TPP needs to add PSU related data, which is requested in the return of the first call. This call might be repeated.
  - The Authorisation Request is only used in an Embedded SCA Approach to authorise the transaction in case a second factor authentication is needed.
  - The Status Request is used e.g. in cases, where the SCA control is taken over by the ASPSP and the TPP needs later information about the outcome.

The following usage of abbreviations in the Location and Usage columns is defined, cp. also [XS2A-OR] for details.

- x: This data element is transported on the corresponding level.
- m: Mandatory
- o: Optional for the TPP to use
- c: Conditional. The condition is described in the addressed API Calls, condition defined by the ASPSP

The following table does not only define requirements on request messages but also requirements on data elements for the response messages. As defined in Section 4.13 these requirements only apply to positive responses (i.e. HTTP response code 2xx). For example, in the case of the Payment Initiation Response Message with HTTP response code 4xx or 5xx, no payment initiation resource has been created and therefore no resource related information can be returned.

Remark: The more technical functions like GET …/{paymentId} and GET …/{authorisationId} and the Cancellation Request are not covered by this table.

Data element	**Attribute encoding**	**Location**					**Usage**							
		Path	Query P.	Header	Body	Certificate[2]	Init Req.	Init Resp.	Upd. Req.	Upd. Resp	Auth. Req.	Auth Resp.	Stat. Req.	Stat. Resp
TPP Registration Number						x	m		m		m		m	
TPP Name						x	m		m		m		m	
TPP Roles						x	m		m		m		m	
TPP National Competent Authority						x	m		m		m		m	
Request Identification	X-Request-ID			x			m	m	m	m	m	m	m	m
Resource ID	paymentId				x			m						

[2] This refers to the certificate for website authentication.

Data element	Attribute encoding	Location					Usage							
		Path	Query P.	Header	Body	Certificate[2]	Init Req.	Init Resp.	Upd. Req.	Upd. Resp	Auth. Req.	Auth Resp.	Stat. Req.	Stat. Resp
Resource ID[3]		x							m		m		m	
Transaction Fees	transactionFees				x			o						
Transaction Fee Indicator	transactionFeeIndicator				x			o						
Access Token (from optional OAuth2)	Authorization			x			c		c		c		c	
Further signature related data	Digest			x			c		c		c		c	
TPP Signing Certificate	TPP-Signature-Certificate			x			c		c		c		c	
TPP Electronic Signature	Signature			x			c		c		c		c	
Transaction Status	transactionStatus				x			m		m		m		m
Funds Availability Flag	fundsAvailable				x									c
PSU Message Information	psuMessage				x			o		o		o		o
TPP Message Information	tppMessages				x			o		o		o		o
PSU Identification	PSU-ID			x			c		c					
PSU Identification Type	PSU-ID-Type			x			c		c					
Corporate Identification	PSU-Corporate-ID			x			c		c		c		c	
Corporate ID Type	PSU-Corporate-ID-Type			x			c		c		c		c	
PSU Password	psuData.password				x				c					

[3] Please note that the Resource ID is transported in the path after the generation of the payment initiation resource. This is then a path parameter without an explicit encoding of the attribute name.

Data element	Attribute encoding	Location					Usage							
		Path	Query P.	Header	Body	Certificate[2]	Init Req.	Init Resp.	Upd. Req.	Upd. Resp	Auth. Req.	Auth Resp.	Stat. Req.	Stat. Resp
Available SCA Methods	scaMethods				x			c		c				
Chosen SCA Method	chosenScaMethod				x				c					
PSU Authentication Data	scaAuthenticationData				x						m			
SCA Challenge Data	challengeData				x			c		c				
IP Address PSU	PSU-IP-Address			x			m		o		o		o	
IP Port PSU	PSU-IP-Port			x			o		o		o		o	
PSU User Agent	PSU-User-Agent[4]			x			o		o		o		o	
GEO Information	PSU-Geo-Location			x			o		o		o		o	
Redirect URL ASPSP	_links.scaRedirect				x			c						
ASPSP-SCA-Approach	ASPSP-SCA-Approach			x				c		c				
Further PSU related Information	PSU-Accept			x			o		o		o		o	
	PSU-Accept-Charset			x			o		o		o		o	
	PSU-Accept-Encoding			x			o		o		o		o	
	PSU-Accept-Language			x			o		o		o		o	
	PSU-Http-Method			x			o		o		o		o	
	PSU-Device-ID			x			o		o		o		o	
Redirect Preference	TPP-Redirect-Preferred			x			o							
Decoupled Preference	TPP-Decoupled-Preferred			x			o							

[4] This field transports key information for risk management like browser type or PSU device operating system. The forwarding of further HTTP header fields might be supported in future versions of the specification to transport other device related information.

Data element	Attribute encoding	Location					Usage							
		Path	Query P.	Header	Body	Certificate[2]	Init Req.	Init Resp.	Upd. Req.	Upd. Resp	Auth. Req.	Auth Resp.	Stat. Req.	Stat. Resp
Redirect URI TPP[5]	TPP-Redirect-URI			x			c							
	TPP-Nok-Redirect_URI			x			o							
Authorisation Preference	TPP-Explicit-Authorisation-Preferred			x			o							
Rejection Preference	TPP-Rejection-NoFunds-Preferred			x			o							
TPP Notification URI	TPP-Notification-URI			x			o							
TPP Notfication Content Preference	TPP-Notification-Content-Preferred			x			o							
TPP Brand Information	TPP-Brand-Logging-Information			x			o							
Payment Product	payment-product	x					m							

The XS2A Interface calls which represent the messages defined in [XS2A-OR] will be defined in the following sections.

> **Remark:** The request timestamp of every call is contained in the mandatory HTTP header "Date", cp. Section 14.37 for the formatting information. This timestamp is not contained in the data tables below because it is a mandatory HTTP header field anyhow and because incompatibilities could appear otherwise with future more formalised specification procedures.
>
> **Remark:** The AIS and PIS service is sharing some sub processes which are once described in Section 7. So, for all Update Data Request/Response Definitions as well as for Authorise Transaction Request/Response Definitions, cp. Section 7.

[5] This redirect link must be contained, if the TPP-Redirect-Preferred flag is contained and equals "true" or if the "TPP-Redirect-Preferred" flag is not used.

PSU IP Address/Port and Further PSU related Information

The above table addresses several PSU related context data. These data, its importance and its usage are defined in detail in Section 4.8. They are not mentioned anymore in the following detailed definitions for matter of better readability, as long as the usage is not mandated.

5.3 Payment Initiation Request

5.3.1 Payment Initiation with JSON encoding of the Payment Instruction

Call

```
POST /v1/payments/{payment-product}
```

Creates a payment initiation request at the ASPSP.

Path Parameters

Attribute	Type	Description
payment-product	String	The addressed payment product endpoint, e.g. for SEPA Credit Transfers (SCT). The default list of products supported in this standard is: • sepa-credit-transfers • instant-sepa-credit-transfers • target-2-payments • cross-border-credit-transfers The ASPSP will publish which of the payment products/endpoints will be supported. For definitions of basic non euro generic products see [XS2A-DP]. Further products might be published by the ASPSP within its XS2A documentation. These new product types will end in further endpoints of the XS2A Interface.

Query Parameters

No Query Parameter

Request Header

Attribute	Type	Condition	Description
Content-Type	String	Mandatory	application/json
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.

Attribute	Type	Condition	Description
			This is the unique ID of TPP for the payment initiation regarding PSD2 article 47 and EBA RTS article 29.
PSU-ID	String	Conditional	Client ID of the PSU in the ASPSP client interface. Might be mandated in the ASPSP's documentation. It might be contained even if an OAuth2 based authentication was performed in a pre-step or an OAuth2 based SCA was performed in a preceding AIS service in the same session. In this case the ASPSP might check whether PSU-ID and token match, according to ASPSP documentation.
PSU-ID-Type	String	Conditional	Type of the PSU-ID; needed in scenarios where PSUs have several PSU-IDs as access possibility. In this case, the mean and use are then defined in the ASPSP's documentation.
PSU-Corporate-ID	String	Conditional	Identification of a Corporate in the Online Channels Might be mandated in the ASPSP's documentation. Only used in a corporate context.
PSU-Corporate-ID-Type	String	Conditional	This is describing the type of the identification needed by the ASPSP to identify the PSU-Corporate-ID content. Mean and use is defined in the ASPSP's documentation. Only used in a corporate context.
Authorization	String	Conditional	Bearer Token. Is contained only, if an OAuth2 based authentication was performed in a pre-step or an OAuth2 based SCA was performed in a preceding AIS service in the same session.
Consent-ID	String	Optional	This data element may be contained, if the payment initiation transaction is part of a session, i.e. combined AIS/PIS service. This then contains the "consentId" of the related AIS consent, which was performed prior to this payment initiation.

Attribute	Type	Condition	Description
PSU-IP-Address	String	Mandatory	The forwarded IP Address header field consists of the corresponding HTTP request IP Address field between PSU and TPP. If not available, the TPP shall use the IP Address used by the TPP when submitting this request.
TPP-Redirect-Preferred	Boolean	Optional	If it equals "true", the TPP prefers a redirect over an embedded SCA approach. If it equals "false", the TPP prefers not to be redirected for SCA. The ASPSP will then choose between the Embedded or the Decoupled SCA approach, depending on the parameter TPP-Decoupled-Preferred and the choice of the SCA procedure by the TPP/PSU. If the parameter is not used, the ASPSP will choose the SCA approach to be applied depending on the SCA method chosen by the TPP/PSU.
TPP-Decoupled-Preferred	Boolean	Optional	If it equals "true", the TPP prefers a decoupled SCA approach. If it equals "false", the TPP prefers not to use the decoupled approach for SCA. The ASPSP will then choose between the embedded or the redirect SCA approach, depending on the choice of the SCA procedure by the TPP/PSU. If the parameter is not used, the ASPSP will choose the SCA approach to be applied depending on the parameter TPP-Redirect-Preferred and the SCA method chosen by the TPP/PSU. The parameter might be ignored by the ASPSP. If both parameters TPP-Redirect-Preferred and TPP-Decoupled-Preferred are present and true, the request is still not rejected, but it is up to the ASPSP, which approach will actually be used.

Attribute	Type	Condition	Description
			RFU: TPP-Redirect-Preferred and TPP-Decoupled-Preferred will be revised in future versions, maybe merged. Currently kept separate for downward compatibility.
TPP-Redirect-URI	String	Conditional	URI of the TPP, where the transaction flow shall be redirected to after a Redirect. Mandated for the Redirect SCA Approach, specifically when TPP-Redirect-Preferred equals "true". See Section 4.10 for further requirements on this header. It is recommended to always use this header field. **Remark for Future**: This field might be changed to mandatory in the next version of the specification.
TPP-Nok-Redirect-URI	String	Optional	If this URI is contained, the TPP is asking to redirect the transaction flow to this address instead of the TPP-Redirect-URI in case of a negative result of the redirect SCA method. This might be ignored by the ASPSP. See Section 4.10 for further requirements on this header.
TPP-Explicit-Authorisation-Preferred	Boolean	Optional	If it equals "true", the TPP prefers to start the authorisation process separately, e.g. because of the usage of a signing basket. This preference might be ignored by the ASPSP, if a signing basket is not supported as functionality. If it equals "false" or if the parameter is not used, there is no preference of the TPP. This especially indicates that the TPP assumes a direct authorisation of the transaction in the next step, without using a signing basket.
TPP-Rejection-NoFunds-Preferred	Boolean	Optional	If it equals "true" then the TPP prefers a rejection of the payment initiation in case the ASPSP is providing an integrated confirmation of funds request an the result of this is that not sufficient funds are available.

Attribute	Type	Condition	Description
			If it equals "false" then the TPP prefers that the ASPSP is dealing with the payment initiation like in the ASPSPs online channel, potentially waiting for a certain time period for funds to arrive to initiate the payment. This parameter may be ignored by the ASPSP.
TPP-Notification-URI	String	Optional	URI for the Endpoint of the TPP-API to which the status of the payment initiation should be sent. This header field **may by ignored** by the ASPSP, cp. also the extended service definition in [XS2A-RSNS].
TPP-Notification-Content-Preferred	String	Optional	The string has the form status=X1, …, Xn where Xi is one of the constants SCA, PROCESS, LAST and where constants are not repeated. The usage of the constants supports the following semantics: SCA: A notification on every change of the scaStatus attribute for all related authorisation processes is preferred by the TPP. PROCESS: A notification on all changes of consentStatus or transactionStatus attributes is preferred by the TPP. LAST: Only a notification on the last consentStatus or transactionStatus as available in the XS2A interface is preferred by the TPP. This header field may be ignored, if the ASPSP does not support resource notification services for the related TPP.
TPP-Brand-Logging-Information	String	Optional	This header might be used by TPPs to inform the ASPSP about the brand used by the TPP towards the PSU. This information is meant for logging

Attribute	Type	Condition	Description
			entries to enhance communication between ASPSP and PSU or ASPSP and TPP. This header might be ignored by the ASPSP.

Remark: Note that a reference of the payment to payer/payee following [PSD2], Article 46 (b), will be handled on application layer with the data attributes related to end2end identification and remittance information, cp. Section 11.1.

Request Body

The payment data to be transported in the request body is dependent on the chosen API endpoint. Some standard definitions related to the above mentioned standard products are defined in Section 11 of this document. Further definitions might be given community or ASPSP specific. In [XS2A-DP], a list of community specific payment product definitions and links regarding community/ASPSP specific payment product definitions are given. ASPSP or community definitions shall reuse standard attribute names.

Response Code

The HTTP response code equals 201.

Response Header

Attribute	Type	Condition	Description
Location	String	Mandatory	Location of the created resource (if created)
X-Request-ID	UUID	Mandatory	ID of the request, unique to the call, as determined by the initiating party.
ASPSP-SCA-Approach	String	Conditional	This data element must be contained, if the SCA Approach is already fixed. Possible values are: • EMBEDDED • DECOUPLED • REDIRECT The OAuth SCA approach will be subsumed by REDIRECT.

Attribute	Type	Condition	Description
ASPSP-Notification-Support	Boolean	Conditional	true if the ASPSP supports resource status notification services. false if the ASPSP supports resource status notification in general, but not for the current request. Not used, if resource status notification services are generally not supported by the ASPSP. Shall be supported if the ASPSP supports resource status notification services, see more details in the extended service definition [XS2A-RSNS].

Attribute	Type	Condition	Description
ASPSP-Notification-Content	String	Conditional	The string has the form status=X1, …, Xn where Xi is one of the constants SCA, PROCESS, LAST and where constants are not repeated. The usage of the constants supports the following semantics: SCA: Notification on every change of the scaStatus attribute for all related authorisation processes is provided by the ASPSP for the related resource. PROCESS: Notification on all changes of consentStatus or transactionStatus attributes is provided by the ASPSP for the related resource. LAST: Notification on the last consentStatus or transactionStatus as available in the XS2A interface is provided by the ASPSP for the related resource. This field must be provided if the ASPSP-Notification-Support =true. The ASPSP might consider the notification content as preferred by the TPP, but can also respond independently of the preferred request.

Response Body

Attribute	Type	Condition	Description
transactionStatus	Transaction Status	Mandatory	The values defined in Section 14.13 might be used.
paymentId	String	Mandatory	resource identification of the generated payment initiation resource.

Attribute	Type	Condition	Description
transactionFees	Amount	Optional	Might be used by the ASPSP to transport the total transaction fee relevant for the underlying payments. This field includes the entry of the currencyConversionFees if applicable.
currency Conversion Fee	Amount	Optional	Might be used by the ASPSP to transport specific currency conversion fees related to the initiated credit transfer.
estimatedTotal Amount	Amount	Optional	The amount which is estimated to be debted from the debtor account. Note: This amount includes fees.
estimated Interbank Settlement Amount	Amount	Optional	The estimated amount to be transferred to the payee.
transactionFee Indicator	Boolean	Optional	If equals true, the transaction will involve specific transaction cost as shown by the ASPSP in their public price list or as agreed between ASPSP and PSU. If equals false, the transaction will not involve additional specific transaction costs to the PSU unless the fee amount is given specifically in the data elements transactionFees and/or currencyConversionFees. If this data element is not used, there is no information about transaction fees unless the fee amount is given explicitly in the data element transactionFees and/or currencyConversionFees.
scaMethods	Array of authentication objects	Conditional	This data element might be contained, if SCA is required and if the PSU has a choice between different authentication methods. Depending on the risk management of the ASPSP this choice might be offered before or after the PSU has been identified with the first relevant factor, or if an access token is transported. If this data element is contained, then there is also a hyperlink of type

Attribute	Type	Condition	Description
			"startAuthorisationWith AuthenticationMethodSelection" contained in the response body. These methods shall be presented towards the PSU for selection by the TPP.
chosenScaMethod	Authentication object	Conditional	This data element is only contained in the response if the ASPSP has chosen the Embedded SCA Approach, if the PSU is already identified e.g. with the first relevant factor or alternatively an access token, if SCA is required and if the authentication method is implicitly selected.
challengeData	Challenge	Conditional	It is contained in addition to the data element "chosenScaMethod" if challenge data is needed for SCA.
			In rare cases this attribute is also used in the context of the "startAuthorisationWith PsuAuthentication" or "startAuthorisactionWithEncryptedPsuAuthentication" link.
_links	Links	Mandatory	A list of hyperlinks to be recognised by the TPP. The actual hyperlinks used in the response depend on the dynamical decisions of the ASPSP when processing the request. **Remark**: All links can be relative or full links, to be decided by the ASPSP. Type of links admitted in this response, (further links might be added for ASPSP defined extensions): "scaRedirect": In case of an SCA Redirect Approach, the ASPSP is transmitting the link to which to redirect the PSU browser. "scaOAuth": In case of a SCA OAuth2 Approach, the ASPSP is transmitting the URI where the configuration of the Authorisation Server can be

Attribute	Type	Condition	Description
			retrieved. The configuration follows the OAuth 2.0 Authorisation Server Metadata specification. "confirmation": Might be added by the ASPSP if either the "scaRedirect" or "scaOAuth" hyperlink is returned in the same response message. This hyperlink defines the URL to the resource which needs to be updated with • a confirmation code as retrieved after the plain redirect authentication process with the ASPSP authentication server or • an access token as retrieved by submitting an authorization code after the integrated OAuth based authentication process with the ASPSP authentication server. "startAuthorisation": In case, where an explicit start of the transaction authorisation is needed, but no more data needs to be updated (no authentication method to be selected, no PSU identification nor PSU authentication data to be uploaded). "startAuthorisationWithPsuIdentification": The link to the authorisation end-point, where the authorisation sub-resource has to be generated while uploading the PSU identification data. "startAuthorisationWithPsuAuthentication": The link to the authorisation end-point, where the authorisation sub-resource has to be generated while uploading the PSU authentication data. "startAuthorisationWithEncryptedPsuAuthentication": Same as startAuthorisactionWithPsuAuthentication, but the

Attribute	Type	Condition	Description
			authentication data need to be encrypted on application level while uploading. "startAuthorisationWithAuthentication MethodSelection": The link to the authorisation end-point, where the authorisation sub-resource has to be generated while selecting the authentication method. This link is contained under exactly the same conditions as the data element "scaMethods" "startAuthorisationWithTransactionAuthorisation": The link to the authorisation end-point, where the authorisation sub-resource has to be generated while authorising the transaction e.g. by uploading an OTP received by SMS. "self": The link to the payment initiation resource created by this request. This link can be used to retrieve the resource data. "status": The link to retrieve the transaction status of the payment initiation. "scaStatus": The link to retrieve the scaStatus of the corresponding authorisation sub-resource. This link is only contained, if an authorisation sub-resource has been already created.
psuMessage	Max500Text	Optional	Text to be displayed to the PSU
tppMessages	Array of TPP Message Information	Optional	Messages to the TPP on operational issues.

Example

Request

```
POST https://api.testbank.com/psd2/v1/payments/sepa-credit-transfers
```

```
Content-Type:           application/json
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
PSU-IP-Address:         192.168.8.78
PSU-GEO-Location:       GEO:52.506931;13.144558
PSU-User-Agent:         Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0
Date:                   Sun, 06 Aug 2017 15:02:37 GMT

{
   "instructedAmount": {"currency": "EUR", "amount": "123.50"},
   "debtorAccount": {"iban": "DE40100100103307118608"},
   "creditorName": "Merchant123",
   "creditorAccount": {"iban": "DE02100100109307118603"},
   "remittanceInformationUnstructured": "Ref Number Merchant"
}
```

Response in case of a redirect with an implicitly created authorisation sub-resource

```
HTTP/1.x 201 Created
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach:     REDIRECT
Date:                   Sun, 06 Aug 2017 15:02:42 GMT
Location:               https://www.testbank.com/psd2/v1/payments/sepa-
credit-transfers/1234-wertiq-983
Content-Type:           application/json

{
  "transactionStatus": "RCVD",
  "paymentId": "1234-wertiq-983",
  "_links": {
        "scaRedirect": {"href": "https://www.testbank.com/asdfasdfasdf"},
        "self": {"href": "/psd2/v1/payments/sepa-credit-transfers/1234-
wertiq-983"},
        "status": {"href": "/psd2/v1/payments/sepa-credit-transfers/1234-
wertiq-983/status"},
        "scaStatus": {"href": "/psd2/v1/payments/sepa-credit-
transfers/1234-wertiq-983/authorisations/123auth456"}
  }
}
```

Same example in case where an explicit authorisation start is needed

```
HTTP/1.x 201 Created
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach:     REDIRECT
```

```
Date:                   Sun, 06 Aug 2017 15:02:42 GMT
Location:               https://www.testbank.com/psd2/v1/payments/sepa-
credit-transfers/1234-wertiq-983
Content-Type:           application/json

{
  "transactionStatus": "RCVD",
  "paymentId": "1234-wertiq-983",
  "_links": {
        "self": {"href": "/psd2/v1/payments/sepa-credit-transfers/1234-
wertiq-983"},
        "status": {"href": "/psd2/v1/payments/sepa-credit-transfers/1234-
wertiq-983/status"},
        "startAuthorisation": {"href": "/psd2/v1/payments/sepa-credit-
transfers/1234-wertiq-983/authorisations"}
  }
}
```

Response in case of an OAuth2 SCA approach with implicitly creating an authorisation sub-resource

```
HTTP/1.x 201 Created
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach:     REDIRECT
Date:                   Sun, 06 Aug 2017 15:02:42 GMT
Location:               https://www.testbank.com/psd2/v1/payments/sepa-
credit-transfers/1234-wertiq-983
Content-Type:           application/json

{
  "transactionStatus": "RCVD",
  "paymentId": "1234-wertiq-983",
  "_links": {
        "scaOAuth": {"href": "https://www.testbank.com/oauth/.well-
known/oauth-authorization-server"},
        "self": {"href": "/psd2/v1/payments/sepa-credit-transfers/1234-
wertiq-983"},
        "status": {"href": "/psd2/v1/payments/sepa-credit-transfers/1234-
wertiq-983/status"},
        "scaStatus": {"href": "/psd2/v1/payments/sepa-credit-
transfers/1234-wertiq-983/authorisations/123auth456"}
  }
}
```

Response in case of the decoupled approach with explicit start of authorisation needed (will be done with the update PSU identification function)

```
HTTP/1.x 201 Created
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach:     DECOUPLED
Date:                   Sun, 06 Aug 2017 15:03:47 GMT
Location:               https://www.testbank.com/psd2/v1/payments/sepa-
credit-transfers/1234-wertiq-983
Content-Type:           application/json

{
  "transactionStatus": "RCVD",
  "paymentId": "1234-wertiq-983",
  "_links": {
     "startAuthorisationWithPsuIdentification": {"href":
"/psd2/v1/payments/sepa-credit-transfers/1234-wertiq-983/authorisations"},
     "self": {"href": "/psd2/v1/payments/sepa-credit-transfers/1234-wertiq-
983"}
    }
}
```

Response in case of the embedded approach with explicit start of authorisation

```
HTTP/1.x 201 Created
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
ASPSP-SCA-Approach:     EMBEDDED
Date:                   Sun, 06 Aug 2017 15:03:47 GMT
Location:               https://www.testbank.com/psd2/v1/payments/sepa-
credit-transfers/1234-wertiq-983
Content-Type:           application/json

{
    "transactionStatus": "RCVD",
    "paymentId": "1234-wertiq-983",
    "_links": {
        "startAuthorisationWithPsuAuthentication": {"href":
"/psd2/v1/payments/sepa-credit-transfers/1234-wertiq-983/authorisations"},
         "self": {"href": "/psd2/v1/payments/sepa-credit-transfers/1234-
wertiq-983"}
               }
}
```

5.3.2 Payment Initiation with pain.001 XML message as Payment Instruction

Call

```
POST /v1/payments/{payment-product}
```

Creates a payment initiation request at the ASPSP.

Remark: The underlying pain.001 structure which is transported in the content body of this request may only contain one payment. In cases of the initiation of bulk payments, the endpoint defined in Section 5.3.3.2 shall be used.

Path Parameters

Attribute	Type	Description
payment-product	String	The addressed payment product, e.g. SCT. The default list of products supported in this standard is: • pain.001-sepa-credit-transfers • pain.001-instant-sepa-credit-transfers • pain.001-target-2-payments • pain.001-cross-border-credit-transfers Further products might be published by the ASPSP within its XS2A documentation. **Remark**: For all SEPA Credit Transfer based endpoints which accept XML encoding, the XML pain.001 schemes provided by EPC are supported by the ASPSP as a minimum for the body content. Further XML schemes might be supported by some communities. **Remark**: For cross-border and target-2 payments only community wide pain.001 schemes do exist, cp. [XS2A-DP].

Query Parameters

The same query parameter definition as in Section 5.3.1 applies.

Request Header

The same header as in Section 5.3.1, only the content type indicates XML encoding ("application/xml").

Request Body

A pain.001 structure corresponding to the chosen payment product, see above on XML schema support.

Response

The same response as in Section 5.3.1.

Example

Request

```
POST https://api.testbank.com/psd2/v1/payments/pain.001-sepa-credit-transfers
Content-Type:           application/xml
X-Request-ID:           "123e4567-e89b-12d3-a456-426655440000"
PSU-IP-Address:         "192.168.8.78"
PSU-User-Agent:         "Chrome_v12"

<Document xmlns="urn:iso:std:iso:20022:tech:xsd:pain.001.001.03">
  <CstmrCdtTrfInitn>
    <GrpHdr>
      <MsgId>MIPI-123456789RI-123456789</MsgId>
      <CreDtTm>2017-02-14T20:23:34.000Z</CreDtTm>
      <NbOfTxs>1</NbOfTxs>
      <CtrlSum>123</CtrlSum>
      <InitgPty>
        <Nm>PaymentInitiator</Nm>
        <Id><OrgId><Othr><Id>DE10000000012</Id>
          <SchmeNm><Prptry>PISP</Prptry></SchmeNm></Othr></OrgId></Id>
      </InitgPty>
    </GrpHdr>
    <PmtInf>
      <PmtInfId>BIPI-123456789RI-123456789</PmtInfId>
      <PmtMtd>TRF</PmtMtd>
      <NbOfTxs>1</NbOfTxs>
      <CtrlSum>123</CtrlSum>
      <PmtTpInf><SvcLvl><Cd>SEPA</Cd></SvcLvl></PmtTpInf>
      <ReqdExctnDt>2017-02-15</ReqdExctnDt>
      <Dbtr><Nm>PSU Name</Nm></Dbtr>
      <DbtrAcct><Id><IBAN>DE87200500001234567890</IBAN></Id></DbtrAcct>
      <ChrgBr>SLEV</ChrgBr>
      <CdtTrfTxInf>
        <PmtId><EndToEndId>RI-123456789</EndToEndId></PmtId>
        <Amt><InstdAmt Ccy="EUR">123</InstdAmt></Amt>
        <Cdtr><Nm>Merchant123</Nm></Cdtr>
        <CdtrAcct><Id><IBAN> DE23100120020123456789</IBAN></Id></CdtrAcct>
        <RmtInf><Ustrd>Ref Number Merchant-123456</Ustrd></RmtInf>
      </CdtTrfTxInf>
    </PmtInf>
  </CstmrCdtTrfInitn>
</Document>
```

Response

See the example responses in JSON encoding in Section 5.3.1

5.3.3 Payment Initiation for Bulk Payments

This function supports the upload of bulk payments. This function is an **optional** function of the ASPSP in the XS2A interface. It can be offered by the ASPSP in JSON or XML modelling of the payment data, i.e. the body content.

5.3.3.1 Bulk Payment Initiation with JSON encoding of the Payment Instruction

Call

```
POST /v1/bulk-payments/{payment-product}
```

Creates a bulk payment initiation request at the ASPSP.

Path Parameters

Attribute	Type	Description
payment-product	String	The addressed payment product endpoint for bulk payments e.g. for a bulk SEPA Credit Transfers (SCT). These endpoints are optional. Some default names are: • sepa-credit-transfers • instant-sepa-credit-transfers • target-2-payments • cross-border-credit-transfers The ASPSP will publish which of the payment products/endpoints will be supported. For definitions of basic non euro generic products see [XS2A-DP]..