| Attribute | Type | Description |
|---|---|---|
| | | 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**

The same query parameter definition as in Section 5.3.1 applies.

**Request Headers**

The same HTTP header definition as in Section 5.3.1 applies.

**Request Body**

The body definition with the JSON based SEPA bulk payments is contained in Section 11.3, further definitions for non SEPA payments in [XS2A-DP]..

**Response**

The responses definition is analogous to the initiation of single payments, cp. Section 5.3.1.

**Remark:** Please note that the optional batchBookingPreferred flag shall be ignored by the ASPSP if batch booking is not supported.

**Recommendations:**

- If the bulk entry contains transactions without a reference, which are rejected, then the whole bulk should be rejected.
- ASPSPs might make the usage of instructionIds or UETR mandatory for bulk initiations.

#### 5.3.3.2 Bulk Payment Initiation with XML 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:<br>• pain.001-sepa-credit-transfers<br>• pain.001-instant-sepa-credit-transfers<br>• pain.001-proprietary-credit-transfers<br>The ASPSP will publish which of the payment products/endpoints will be supported.<br>**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.<br>**Remark**: Payment Initiations might be further restricted by the ASPSP on size or on multiplicity of entries. This could be e.g. a restriction on the usage of one ordering party or/and one debtor account.<br>**Remark**: For proprietary payments, only community wide pain.001 schemes do exist, [XS2A-DP]. |

**Query Parameters**

The same query parameter definition as in Section 5.3.2 applies.

**Request Headers**

The same HTTP header definition as in Section 5.3.2 applies

**Request Body**

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

**Response**

The responses definition is analogous to the initiation of single XML based payments, cp Section 5.3.2.

### 5.3.4 Initiation for Standing Orders for Recurring/Periodic Payments

The recurring payments initiation function will be covered in this specification as a specific standing order initiation: The TPP can submit a recurring payment initiation where the starting date, frequency and conditionally an end date is provided. Once authorised by the PSU, the payment then will be executed by the ASPSP, if possible, following this "standing order" as submitted by the TPP. No further TPP action is needed. This payment is called a periodic payment in this context to differentiate the payment from recurring payment types, where third parties are initiating the same amount of money e.g. payees for using credit card transactions or direct debits for recurring payments of goods or services. These latter types of payment initiations are not part of this interface.

#### 5.3.4.1 Standing Orders for Recurring/Periodic Payments in JSON encoding

**Call**

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

**Path Parameters**

The same path parameter to determine the underlying payment type of the recurring payment as in Section 5.3.1 applies.

**Query Parameters**

The same query parameter definition as in Section 5.3.1 applies.

**Request Header**

For this initiation the same header as in Section 5.3.1 is used.

**Request Body**

First, any tag of the underlying payment as defined in Section 11.1 can be used. In addition the following tags are used:

| Tag | Type | Usage | Description |
|---|---|---|---|
| startDate | ISODate | {Or | The first applicable day of execution starting from this date is the first payment. |
| startDateTime | ISODateTime | Or} | The first applicable day and time of execution starting from this timestamp.<br>This attribute is only applicable to instant payments. Seconds and milli |

| Tag | Type | Usage | Description |
|---|---|---|---|
| | | | seconds might be ignored by the ASPSP.<br>The ASPSP might restrict the implementation to startDate. This restriction will be part of the ASPSP documentation. |
| executionRule | String | Optional | "following" or "preceding" supported as values. This data attribute defines the behavior when recurring payment dates falls on a weekend or bank holiday. The payment is then executed either the "preceding" or "following" working day.<br>ASPSP might reject the request due to the communicated value, if rules in Online-Banking are not supporting this execution rule. |
| endDate | ISODate | {Or Optional | The last applicable day of execution<br>If none of the attributes is given, it is an infinite standing order. |
| endDateTime | ISODateTime | Optional Or} | The last applicable day and time of requested execution.<br>This attribute is only applicable to instant payments.<br>If none of the attributes is given, it is an infinite standing order.<br>The ASPSP might restrict the implementation to endDate. This restriction will be part of the ASPSP documentation. |
| frequency | Frequency Code | Mandatory | The frequency of the recurring payment resulting from this standing order. |

| Tag | Type | Usage | Description |
|---|---|---|---|
| dayOfExecution | Max2Text | Conditional | "31" is ultimo.<br>The format is following the regular expression \d{1,2}.<br>Example: The first day is addressed by "1".<br>The date is referring to the time zone of the ASPSP. |
| monthsOfExecution | Array of Max2Text | Conditional | The format is following the regular expression \d{1,2}. The array is restricted to 11 entries. The values contained In the array entries shall all be different and the maximum value of one entry is 12.<br>This attribute is contained if and only if the frequency equals "MonthlyVariable".<br>Example: An execution on January, April and October each year is addressed by ["1". "4", "10"]. |

**Response**

The formats of the Payment Initiation Response resp. the subsequent transaction authorisation process for standing orders with JSON based payment data equals the corresponding Payment Initiation Response resp. the subsequent transaction authorisation process for a single payment containing JSON based payment data.

> **Remark**: Please note that for the payment initiation of standing orders, the ASPSP will always mandate an SCA with dynamic linking, exemptions are not permitted.

**Example**

***Request for Variant 1 with full JSON encoding***

```
POST https://www.testbank.com/psd2/v1/periodic-payments/sepa-credit-
transfers
Content-Type:           application/json
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
PSU-IP-Address:         192.168.8.78
```

```
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"},
 "debtorAccount": {"iban": "DE40100100103307118608"},
 "creditorName": "Merchant123",
 "creditorAccount": {"iban": "DE23100120020123456789"},
 "remittanceInformationUnstructured": "Ref Number Abonnement",
 "startDate": "2018-03-01",
 "executionRule": "preceding",
 "frequency": "Monthly",
 "dayOfExecution": "01"
}
```

#### 5.3.4.2 Payment Initiation for Standing Orders with XML based payment data

The standing order management data will be JSON based in the XS2A API also if the related payment data is based on XML syntax. For this reason, the Payment Initiation Request for standing orders is defined as an HTTP multipart message in this case.

**Call**

```
POST /v1/periodic-payments/{product-name}
```

**Path Parameters**

The same path parameter to determine the underlying payment type of the recurring payment as in Section 5.3.2 applies.

**Query Parameters**

The same query parameter and HTTP header definition as in Section 5.3.1 applies.

**Request Header**

The same header definitions as in Section 5.3.1 are used with the exception of the Content-Type Header. Here the following requirement applies:

| Attribute | Type | Condition | Description |
|---|---|---|---|
| Content-Type | String | Mandatory | multipart/form-data; boundary=AaaBbbCcc |

**Request Body, Part 1**

The first part of the body contains first a sub-header section as defined by the following table:

| Attribute | Type | Condition | Description |
|---|---|---|---|
| Content-Disposition | String | Mandatory | form-data; name="xml_sct" |
| Content-Type | String | Mandatory | application/xml |

The first part content of the body is defined as for the Payment Initiation Request for a single request in an XML (pain.001) based format, cp. Section 5.3.2.

**Request Body, Part 2**

The second part of the body contains first a sub-header section as defined by the following table:

| Attribute | Type | Condition | Description |
|---|---|---|---|
| Content-Disposition | String | Mandatory | form-data; name="json_standingorderType" |
| Content-Type | String | Mandatory | application/json |

The second part content of the body is defined as follows:

| Tag | Type | Usage | Description |
|---|---|---|---|
| startDate | ISODate | Mandatory | The first applicable day of execution starting from this date is the first payment. |
| executionRule | String | Optional | "following" or "preceding" supported as values. This data attribute defines the behavior when recurring payment dates falls on a weekend or bank holiday. The payment is then executed either the "preceding" or "following" working day.ASPSP might reject the request due to the communicated value, if rules in Online-Banking are not supporting this execution rule. |

| Tag | Type | Usage | Description |
|---|---|---|---|
| endDate | ISODate | Optional | The last applicable day of execution<br>If not given, it is an infinite standing order. |
| frequency | Frequency Code | Mandatory | Frequency of the recurring payment resulting from this standing order. |
| dayOfExecution | Max2Text | Conditional | "31" is ultimo |
| monthsOfExecution | Array of Max2Text | Conditional | The format is following the regular expression \d{1,2}. The array is restricted to 11 entries. The values contained In the array entries shall all be different and the maximum value of one entry is 12.<br>This attribute is contained if and only if the frequency equals "MonthlyVariable".<br>Example: An execution on January, April and October each year is addressed by ["1". "4", "10"]. |

### Response

The formats of the Payment Initiation Response resp. the subsequent transaction authorisation process for standing orders with XML based payment data equals the corresponding Payment Initiation Response resp. the subsequent transaction authorisation process for a single payment containing XML based payment data.

### Example

### Request with JSON Management Information and XML Payment Information

```
POST https://www.testbank.com/psd2/v1/periodic-payments/sepa-credit-
transfers
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
Content-Type: multipart/form-data; boundary=AaaBbbCcc
```

```
--AaaBbbCcc
Content-Disposition: form-data; name="xml_sct"
Content-Type: application/xml
<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>
--AaaBbbCcc
Content-Disposition: form-data; name="json_standingordermanagement"
Content-Type: application/json
{"startDate": "2018-03-01",
 "frequency": "Monthly",
 "executionRule": "preceding",
 "dayOfExecution": "01"
}
--AaaBbbCcc—-
```

## 5.4 Get Transaction Status Request

### Call

```
GET /v1/{payment-service}/{payment-product}/{paymentId}/status
```

Can check the status of a payment initiation.

### Path Parameter

| Attribute | Type | Description |
|---|---|---|
| payment-service | String | The possible values are "payments", "bulk-payments" and "periodic-payments" |
| payment-product | String | The payment product, under which the payment under paymentId has been initiated.<br>It shall be checked by the ASPSP, if the payment-product is matching the payment initiation addressed by paymentId. |
| paymentId | String | Resource Identification of the related payment. |

### Request Header

| Attribute | Type | Condition | Description |
|---|---|---|---|
| X-Request-ID | UUID | Mandatory | ID of the request, unique to the call, as determined by the initiating party. |
| Authorization | String | Conditional | Is contained only, if an OAuth2 based authentication was performed in a pre-step or an OAuth2 based SCA was performed in the current PIS transaction or in a preceding AIS service in the same session, if no such OAuth2 SCA approach was chosen in the current PIS transaction. |

| Attribute | Type | Condition | Description |
|---|---|---|---|
| Accept | String | Optional | The TPP can indicate the formats of status reports supported together with a prioritisation following the HTTP header definition.<br>The formats supported by this specification are<br>• xml<br>• JSON<br>If only one format is supported by the TPP, which is not supported by the ASPSP this can lead to a rejection of the request. |

### Query Parameters

No specific query parameters defined.

### Request Body

No request body.

### Response Code

The HTTP response code equals 200.

### Response Header

| Attribute | Type | Condition | Description |
|---|---|---|---|
| X-Request-ID | UUID | Mandatory | ID of the request, unique to the call, as determined by the initiating party. |

### Response Body in Case of JSON based endpoint

| Attribute | Type | Condition | Description |
|---|---|---|---|
| transactionStatus | Transaction Status | Mandatory | In case where the Payment Initiation Request was JSON encoded as defined in Section 5.3.1, the status is returned in this JSON based encoding.<br>**Remark:** If the PSU does not complete a required SCA within the required timeframe the payment resource's status must be set to "RJCT". Particularly, if a multi-level-SCA is required and the number of successful SCAs |

| Attribute | Type | Condition | Description |
|---|---|---|---|
| | | | during the required timeframe is insufficient, the status must also be set to "RJCT". |
| fundsAvailable | Boolean | Conditional | This data element is contained, if supported by the ASPSP, if a funds check has been performed and if the transactionStatus is "ACTC", "ACWC" or "ACCP". |
| psuMessage | Max500Text | Optional | |
| ownerNames | Array of Account Owner | Optional | List of owner names.<br>Should only be delivered after successful SCA. Could be restricted to the current PSU by the ASPSP. |
| psuName | Max140Text | Optional | Name of the PSU[6]<br>In case of a corporate account, this might be the person acting on behalf of the corporate. |
| _links | Links | Optional | Should refer to next steps if the problem can be resolved via the interface e.g. for re-submission of credentials. |
| tppMessages | Array of TPP Message Information | Optional | Messages to the TPP on operational issues. |

### Response Body in Case of (SEPA-)XML based endpoint

If the Payment Initiation Request is encoded in XML, cp. Section 5.3.2, then the status might be returned by the ASPSP as a pain.002 structure or as JSON structure as defined above. The ASPSP can choose in this case one of the two status formats or offer both. In case of an XML format, the chosen XML schema of the Status Request is following the XML schema definitions of the original pain.001 schema.

[6] Usage is following the mandate resulting from EBA Q&A 2020_5165

## Example

### *Example for JSON based endpoint*

### *Request*

```
GET https://api.testbank.com/psd2/v1/payments/1234-wertiq-983/status
Accept:                 application/json
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                   Sun, 06 Aug 2017 15:04:07 GMT
```

### *Response*

```
HTTP/1.x 200 Ok
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                   Sun, 06 Aug 2017 15:04:08 GMT
Content-Type:           application/json

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

### *Example for XML based endpoint*

### *Request*

```
GET https://api.testbank.com/psd2/v1/payments/pain.001-sepa-credit-
transfers/1234-wertiq-983/status
Accept:                 application/xml, application/json;q=0.9
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                   Sun, 06 Aug 2017 15:04:07 GMT
```

### *Response*

```
HTTP/1.x 200 Ok
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                   Sun, 06 Aug 2017 15:04:08 GMT
Content-Type:           application/xml

<Document xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.03">
..<CstmrPmtStsRpt>
….<GrpHdr>
…...<MsgId>4572457256725689726906</MsgId>
......<CreDtTm>2017-02-14T20:24:56.021Z</CreDtTm>
......<DbtrAgt><FinInstnId><BIC>ABCDDEFF</BIC></FinInstnId></DbtrAgt>
```

```
......<CdtrAgt><FinInstnId><BIC>DCBADEFF</BIC></FinInstnId></CdtrAgt>
....</GrpHdr>
....<OrgnlGrpInfAndSts>
......<OrgnlMsgId>MIPI-123456789RI-123456789</OrgnlMsgId>
......<OrgnlMsgNmId>pain.001.001.03</OrgnlMsgNmId>
......<OrgnlCreDtTm>2017-02-14T20:23:34.000Z</OrgnlCreDtTm>
......<OrgnlNbOfTxs>1</OrgnlNbOfTxs>
......<OrgnlCtrlSum>123</OrgnlCtrlSum>
......<GrpSts>ACCT</GrpSts>
....</OrgnlGrpInfAndSts>
....<OrgnlPmtInfAndSts>
......<OrgnlPmtInfId>BIPI-123456789RI-123456789</OrgnlPmtInfId>
......<OrgnlNbOfTxs>1</OrgnlNbOfTxs>
......<OrgnlCtrlSum>123</OrgnlCtrlSum>
......<PmtInfSts>ACCT</PmtInfSts>
....</OrgnlPmtInfAndSts>
..</CstmrPmtStsRpt>
</Document>
```

## 5.5 Get Bulk Extended Status Request

### Call

```
GET /v1/bulk-payments/{payment-product}/{paymentId}/extended-status
```

Can check the overall status of a bulk payment initiation together with reporting rejected transactions.

### Path Parameter

| Attribute | Type | Description |
|---|---|---|
| payment-product | String | The payment product, under which the payment under paymentId has been initiated. The standardised payment products covered for this service are<br>- sepa-credit-transfers<br>- instant-sepa-credit-transfers<br>- crossborder-payments<br>**NOTE**: This report is not supported for XML based bulk payments. |
| paymentId | String | Resource Identification of the related bulk payment. |

### Request Header

| Attribute | Type | Condition | Description |
|---|---|---|---|
| X-Request-ID | UUID | Mandatory | ID of the request, unique to the call, as determined by the initiating party. |
| Authorization | String | Conditional | Is contained only, if an OAuth2 based authentication was performed in a pre-step or an OAuth2 based SCA was performed in the current PIS transaction or in a preceding AIS service in the same session, if no such OAuth2 SCA approach was chosen in the current PIS transaction. |

### Query Parameters

No specific query parameters defined.

### Request Body

No request body.

### Response Code

The HTTP response code equals 200.

### Response Header

| Attribute | Type | Condition | Description |
|---|---|---|---|
| X-Request-ID | UUID | Mandatory | ID of the request, unique to the call, as determined by the initiating party. |

### Response Body

| Attribute | Type | Condition | Description |
|---|---|---|---|
| groupStatus | Group Status | Mandatory | This is the bulk status as defined in the related bulk status endpoint. |

| Attribute | Type | Condition | Description |
|---|---|---|---|
| reasonCode | Status Reason Code | {Or - Optional | Additional information on the reason for e.g. rejecting the request |
| reasonProprietary | Max35Text | Or – Optional} | Proprietary additional information on the reason for e.g. rejecting the request.<br>Remark: If an ISO Code is available it should be used instead of Proprietary Reasons. Further restrictions could be defined by API Access Schemes. |
| fundsAvailable | Boolean | Conditional | This data element is contained, if supported by the ASPSP, if a funds check has been performed and if the transactionStatus is "ACTC", "ACWC" or "ACCP". |
| originalTransaction InformationAndStatus | Array of Original Transaction Information And Status | Conditional | Provides information on the original transactions to which the status report message refers and which have been rejected. |
| transactionFees | Amount | Optional | Might be used by the ASPSP to transport the total transaction fee relevant in case of an underlying payment initiation service. This field includes the entry of the currencyConversionFees if applicable. |
| currencyConversion Fees | Amount | Optional | Might be used by the ASPSP to transport specific currency conversion fees related to the initiated credit transfer, in case of an underlying payment initiation service. |
| estimatedTotalAmount | Amount | Optional | The amount which is estimated to be debted from the debtor account in case of an underlying payment initiation service.<br>Note: This amount includes fees. |

| Attribute | Type | Condition | Description |
|---|---|---|---|
| estimatedInterbank SettlementAmount | Amount | Optional | The estimated amount to be transferred to the payee in case of an underlying payment initiation service. |
| psuMessage | Max500Text | Optional | |
| ownerNames | Array of Account Owner | Optional | List of owner names.<br>Should only be delivered after successful SCA. Could be restricted to the current PSU by the ASPSP. |
| _links | Links | Optional | Should refer to next steps if a potential situation blocking the finalisation of the payment can be resolved via the interface e.g. for re-submission of credentials. |
| apiClientMessages | Array of API Client Message Information | Optional | Messages to the TPP on operational issues. |

### Example

#### *Request*

```
GET https://api.testbank.com/psd2/v1/bulk-payments/sepa-credit-transfers/
3d9a81b3-a47d-4130-8765-a9c0ff861100/extended-status
Accept:                 application/json
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                   Sun, 06 Aug 2017 15:04:07 GMT
```

#### *Response*

```
HTTP/1.x 200 Ok
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                   Sun, 06 Aug 2017 15:04:08 GMT
Content-Type:           application/json

{
  "groupStatus": "PART",
  "originalTransactionInformationAndStatus": [
    {
      "originalUetr": "de2da6c9-18be-48d4-8053-867ed90a316a",
      "transactionStatus": "RJCT"
    },
    {
      "originalUetr": "de2da6c9-18be-48d4-8053-867ed90a317a",
      "transactionStatus": "RJCT"
    }
  ]
}
```

## 5.6 Get Payment Request

```
GET /v1/{payment-service}/{payment-product}/{paymentId}
```

Returns the content of a payment object.

### Path Parameters

| Attribute | Type | Description |
|---|---|---|
| payment-service | String | The possible values are “payments”, “bulk-payments” and “periodic-payments” |
| payment-product | String | The payment product, under which the payment under paymentId has been initiated. |
| paymentId | String | ID of the corresponding payment initiation object as returned by an Payment Initiation Request |

### Query Parameters

No specific query parameter.

### Request Headers

| Attribute | Type | Condition | Description |
|---|---|---|---|
| X-Request-ID | UUID | Mandatory | ID of the request, unique to the call, as determined by the initiating party. |

| Attribute | Type | Condition | Description |
|---|---|---|---|
| Authorization | String | Conditional | Is contained only, if an OAuth2 based authentication was performed in a pre-step or an OAuth2 based SCA was performed in the current PIS transaction or in a preceding AIS service in the same session, if no such OAuth2 SCA approach was chosen in the current PIS transaction. |

**Request Body**

No request body.

**Response Header**

| Attribute | Type | Condition | Description |
|---|---|---|---|
| X-Request-ID | UUID | Mandatory | ID of the request, unique to the call, as determined by the initiating party. |

**Response Code**

The HTTP response code equals 200.

**Response Body**

The response body is dependent on the parameter {payment-service}. It contains the view of the ASPSP on the addressed payment resource.

For JSON based {payment-services}, the payment resources may contain e.g. in addition the transaction status data element.

**Note:** In addition, the payment resource may contain the debtorName field even if it was not provided by the TPP. This enables the ASPSP to transport the account owner name to the PISP in case where the regulatory need is provided and if not provided by other means like the List of Available Accounts Service or general AIS services for AISPs.

**Note:** According to item 40 of [EBA-OP2] the payment resource shall contain the debtorAccount after the payment has been initiated successfully, even if it was not provided by the TPP within the initial call.

For XML based {payment-services}, the pain.001 objects are returned. In case of a submitted standing order where the payment information has been submitted in a pain.001 format, the resource content is returned in a multipart message as the submission.

In all cases, the data element entries can be different from the submission entries, if the ASPSP has reformatted the content, e.g. the requested execution dates or character sets in the unstructured remittance information.

## 5.7 Payment Cancellation Request

### Call

```
DELETE /v1/{payment-service}/{payment-product}/{paymentId}
```

It initiates the cancellation of a payment. Depending on the payment-service, the payment-product and the ASPSP's implementation, this TPP call might be sufficient to cancel a payment. If an authorisation of the payment cancellation is mandated by the ASPSP, a corresponding hyperlink will be contained in the response message. These two cases will be separated also in using different 2xx HTTP response codes.

### Path Parameter

| Attribute | Type | Description |
|---|---|---|
| payment-service | String | The possible values are “payments”, “bulk-payments” and “periodic-payments” |
| payment-product | String | The payment product, under which the payment under paymentId has been initiated.<br>It shall be checked by the ASPSP, if the payment-product is matching the payment initiation addressed by paymentId. |
| paymentId | String | Resource Identification of the related payment. |

### Request Header

| Attribute | Type | Condition | Description |
|---|---|---|---|
| X-Request-ID | UUID | Mandatory | ID of the request, unique to the call, as determined by the initiating party. |

| Attribute | Type | Condition | Description |
|---|---|---|---|
| Authorization | String | Conditional | Is contained only, if an OAuth2 based authentication was performed in a pre-step or an OAuth2 based SCA was performed in the current PIS transaction or in a preceding AIS service in the same session, if no such OAuth2 SCA approach was chosen in the current PIS transaction. |
| TPP-Redirect-Preferred | Boolean | Optional | If it equals "true", the TPP prefers a redirect over an embedded SCA approach.<br>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.<br>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. |

| Attribute | Type | Condition | Description |
|---|---|---|---|
| TPP-Decoupled-Preferred | Boolean | Optional | If it equals "true", the TPP prefers a decoupled SCA approach.<br>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.<br>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.<br>The parameter might be ignored by the ASPSP.<br>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.<br>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.<br>It is recommended to always use this header field.<br>**Remark for Future**: This field might be changed to mandatory in the next version of the specification. |

| Attribute | Type | Condition | Description |
|---|---|---|---|
| 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.<br>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.<br>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. |

### Query Parameters

No specific query parameters defined.

### Request Body

No request body.

### Response Code

If the DELETE is sufficient for cancelling the payment: HTTP response code 204.

If the DELETE is not sufficient for cancelling the payment since an authorisation of the cancellation by the PSU is needed: HTTP response code 202.

### Response Header

| Attribute | Type | Condition | Description |
|---|---|---|---|
| X-Request-ID | UUID | Mandatory | ID of the request, unique to the call, as determined by the initiating party. |

### Response Body

In case of HTTP code 204, no response body is used.

In case of HTTP code 202, the following body is used:

| Attribute | Type | Condition | Description |
|---|---|---|---|
| transactionStatus | Transaction Status | Mandatory | Transaction Status of the payment resource |
| 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 "startAuthorisationWith AuthenticationMethodsSelection" contained in the response body.<br>These methods shall be presented towards the PSU for selection by the TPP. |
| chosenSca Method | 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 |

| Attribute | Type | Condition | Description |
|---|---|---|---|
| | | | PsuAuthentication" or "startAuthorisactionWith EncryptedPsuAuthentication" link. |
| _links | Links | Conditional | 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.<br>**Remark**: All links can be relative or full links, to be decided by the ASPSP.<br>Type of links admitted in this response, (further links might be added for ASPSP defined extensions): |
| | | | "startAuthorisation":<br>In case, where just the authorisation process of the cancellation needs to be started, but no additional data needs to be updated for time being (no authentication method to be selected, no PSU identification nor PSU authentication data to be uploaded). |
| | | | "startAuthorisationWithPsuIdentification":<br>In case where a PSU Identification needs to be updated when starting the cancellation authorisation: The link to the cancellation-authorisations end-point, where the cancellation sub-resource has to be generated while uploading the PSU identification data. |
| | | | "startAuthorisationWithPsuAuthentication":<br>In case of a yet to be created authorisation sub-resource: The link to the cancellation-authorisation end-point, where the authorisation sub-resource has to be generated while uploading the PSU authentication data. |
| | | | "startAuthorisationWithEncryptedPsuAuthentication": |

| Attribute | Type | Condition | Description |
|---|---|---|---|
| | | | Same as startAuthorisactionWithPsu Authentication where the authentication data need to be encrypted on application layer in uploading. |
| | | | "startAuthorisationWithAuthentication MethodSelection":<br>The link to the authorisation end-point, where the cancellation-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" |

### *Example in case the DELETE process as such is already sufficient for cancelling the payment*

### *Request*

```
DELETE https://api.testbank.com/psd2/v1/payments/sepa-credit-transfers/123456scheduled789

Content-Type            application/json
X-Request-ID            99391c7e-ad88-49ec-a2ad-99ddcb1f7769
Date                    Sun, 13 Aug 2017 17:05:37 GMT
```

### *Response*

```
HTTP/1.x 204
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7769
Date:                   Sun, 13 Aug 2017 17:05:38 GMT
```

### *Example in case an authorisation of the cancellation is needed by the PSU*

### *Request*

### *Request*

```
DELETE https://api.testbank.com/psd2/v1/payments/sepa-credit-transfers/123456scheduled789
```

```
Content-Type           application/json
X-Request-ID           99391c7e-ad88-49ec-a2ad-99ddcb1f7769
Date                   Sun, 13 Aug 2017 17:05:37 GMT
```

***Response***

```
HTTP/1.x 202
X-Request-ID:          99391c7e-ad88-49ec-a2ad-99ddcb1f7769
Date:                  Sun, 13 Aug 2017 17:05:38 GMT
{"transactionStatus": "ACTC",
"_links": {
        "self": {"href": "/psd2/v1/payments/sepa-credit-
transfers/123456scheduled789"},
        "status": {"href": "/psd2/v1/payments/sepa-credit-
transfers/123456scheduled789/status"},
        "startAuthorisation": {"href": "/psd2/v1/payments/sepa-credit-
transfers/123456scheduled789/cancellation-authorisations"}
  }
}
```

## 5.8 Get Cancellation Authorisation Sub-Resources Request

### Call in context of a Payment Cancellation Request

```
GET /v1/{payment-service}/{payment-product}/{paymentId}/cancellation-
authorisations
```

Will deliver an array of resource identifications to all generated cancellation authorisation sub-resources.

### Path Parameters

| Attribute | Type | Description |
|---|---|---|
| payment-service | String | The possible values are “payments”, “bulk-payments” and “periodic-payments” |
| payment-product | String | The payment product, under which the payment under paymentId has been initiated.<br>It shall be checked by the ASPSP, if the payment-product is matching the payment initiation addressed by paymentId. |

| Attribute | Type | Description |
|---|---|---|
| paymentId | String | Resource identification of the related payment initiation resource. |

### Query Parameters

No specific query parameters defined.

### Request Header

| Attribute | Type | Condition | Description |
|---|---|---|---|
| X-Request-ID | UUID | Mandatory | ID of the request, unique to the call, as determined by the initiating party. |
| Authorization | String | Conditional | Is contained only, if an OAuth2 based authentication was performed in a pre-step or an OAuth2 based SCA was performed in the current PIS transaction or in a preceding AIS service in the same session, if no such OAuth2 SCA approach was chosen in the current PIS transaction. |

### Request Body

No request body.

### Response Code

The HTTP response code equals 200.

### Response Header

| Attribute | Type | Condition | Description |
|---|---|---|---|
| X-Request-ID | UUID | Mandatory | ID of the request, unique to the call, as determined by the initiating party. |

### Response Body

| Attribute | Type | Condition | Description |
|---|---|---|---|
| authorisationIds | Array of String | Mandatory | An array of all authorisationIds connected to the cancellation of this payment resource. |

### *Example*

### *Request*

```
GET https://api.testbank.com/psd2/v1/payments/sepa-credit-transfers/1234-
wertiq-983/cancellation-authorisations
Accept:                 application/json
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7723
Date:                   Sun, 06 Aug 2017 15:04:07 GMT
```

### *Response*

```
HTTP/1.x 200 Ok
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7723
Date:                   Sun, 06 Aug 2017 15:04:08 GMT
Content-Type:           application/json

{
  "authorisationIds": ["123auth456"]
}
```

## 5.9 Multilevel SCA for Payments

The Payment Initiation Requests defined in this section are independent from the need of one or several SCA processes, i.e. independent from the number of authorisations needed for the execution of payments. In contrast, the Initiation Response messages defined above in this section are specific to the processing of one SCA.. In the following the background is explained on diverging requirements on the Payment Initiation Response messages.

For payment initiation with multilevel SCA, this specification requires an explicit start of the authorisation, i.e. links directly associated with SCA processing like "scaRedirect" or "scaOAuth" cannot be contained in the response message of a Payment Initiation Request for a payment, where multiple authorisations are needed. Also if any data is needed for the next action, like selecting an SCA method is not supported in the response, since all starts of the multiple authorisations are fully equal. In these cases, first an authorisation sub-resource has to be generated following the "startAuthorisation" link.

**Response Body for Payment Initiation Messages with Multilevel SCA**

| 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. |
| transactionFees | Amount | Optional | Can be used by the ASPSP to transport transaction fees relevant for the underlying payments. |
| transactionFeeIndicator | 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.<br>If equals false, the transaction will not involve additional specific transaction costs to the PSU. |
| _links | Links | Mandatory | "startAuthorisation":<br>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).<br>"startAuthorisationWithPsuIdentification":<br>The link to the authorisation end-point, where the authorisation sub-resource has to be generated while uploading the PSU identification data.<br>"startAuthorisationWithPsuAuthentication":<br>The link to the authorisation end-point, where an authorisation sub-resource has to be generated while uploading the PSU authentication data. |

| Attribute | Type | Condition | Description |
|---|---|---|---|
| | | | "startAuthorisationWithEncryptedPsuAuthentication":<br>The link to the authorisation end-point, where an authorisation sub-resource has to be generated while uploading the encrypted PSU authentication data.<br>"self": The link to the payment initiation resource created by this request. This link can be used to retrieve the resource data.<br>"status": The link to retrieve the transaction status of the payment initiation. |
| psuMessage | Max500Text | Optional | Text to be displayed to the PSU |
| tppMessages | Array of TPP Message Information | Optional | Messages to the TPP on operational issues. |

**Remark:** In difference to the Payment Initiation Flow with one SCA, optimisation processes with implicitly generating authorisation sub-resources are not supported for Multiple SCA to keep the several authorisation processes of different PSUs for the same payment identical, so that the start of the authorisation process is context free. That is, the only steering hyperlinks returned to the TPP after a payment initiations are "start authorisation" hyperlinks with information in addition about mandatory data to be uploaded with the Start Authorisation Request (PSU Identification or PSU Authentication data). It is not possible to upload with the first command the selected authentication method or OTP Response data because this would require to transport the selected authentication methods or challenge data before.

## 5.10 Payment Initiation Specifics for Multi-currency Accounts

The payment data contained in the request body can also address sub-accounts which are provided in specific currencies, cp. definition of multi-currency accounts in Section 4.5. This is independent of the coding in JSON or XML.

# 6 Account Information Service

## Supported Sub-Services

This specification foresees different types of account information services:

- Transaction reports for a given account with transactions with booking status booked or pending including balances if applicable,

- List of standing orders of a given account, reported as transactions with booking status information,

- Balances of a given account,

- A list of available accounts,

- Account details of a given account or of the list of all accessible accounts relative to a granted consent, and

- Account details might include the account owner name, where specific requirements on the consent process might apply, see below.

Hereby the definition of the list of available and accessible accounts is as follows:

**Definition:** The list of **available** accounts of an ASPSP related to a PSU is the list of accounts of a PSU which are open for access through the XS2A interface according to the definition of payment accounts provided by [PSD2].

**Definition:** The list of **accessible** accounts of an ASPSP related to a PSU's consent is the list of accounts, where the consent of the PSU has been granted to at least one of the defined account information types.

**Note:** The Read Data Request for the list of available accounts and for account details of a given account is syntactically identical. The difference is only in the underlying consent resource, referred to through the HTTP header parameter "Consent-ID".

**Example:** An ASPSP is providing IBAN1 and IBAN2 to a PSU. The PSU has granted the TPP the consent to access transactions and balances of IBAN1. In this case, the available accounts are IBAN1 and IBAN2, the list of accessible accounts consists only of IBAN1.

## Establishing Consent and Reading Account Data

Within this specification, the Account Information Service is separated in two phases:

- Establish Account Information Consent

  Within this phase of the Account Information Service, the PSU is giving the consent to the AISP on

  - the type of Account Information Service to grant an access to (see list at the beginning of this section),
  - the multiplicity of the Account Information Service, i.e. a one-off or recurring access, and
  - in the latter case on the duration of the consent in days or the maximum offered by the ASPSP and optionally the frequency of a recurring request.

This consent is then authorised by the PSU towards the ASPSP with the SCA as mandated by [EBA-RTS].

The result of this process is a consent resource. A link to this resource is returned to the AISP within this process. The TPP can retrieve the consent object by submitting a GET method on this resource. This object contains a.o. the detailed access rights, the current validity and a Consent-ID token.

- Read Account Data

  Within this phase, the AISP gets access to the account data as defined by the PSU's consent, see above. The Read Account Data Request is addressing the corresponding consent resource by using the above mentioned link to this resource.

  The Read Account Data Request will indicate

  - the type of account data to be accessed,
  - the identification of the addressed account, where applicable,
  - whether a PSU has directly initiated the request real-time,
  - whether balances should be delivered in addition where applicable,
  - in case of transaction reports as Account Information type additionally
    - the addressed account identification and
    - the period of the transaction report
    - in addition optionally a delta-flag indicating the request for a delta-report relative to the last request with additional data.
    - the preferred formats of the transaction reports.

For the account access, the usual bank accounts and (credit) card accounts are separated on end-points, since the data is usually separated in the ASPSP backend.

In case of a one-off consent, the access might be denied if the AISP is requesting the data more than once or if the validity of the consent has been timed out, e.g. after 20 minutes of the finalisation of the consent mechanism, depending on the ASPSP implementation.

The read data access will be further denied in case where the type of Account Information Service does not comply with the consented service, or if the actual access is not matching the consented duration or frequency.

If the PSU's consent is given to access a list of accounts, the frequency of the access is checked by the ASPSP per account that has been accessed and per PSU that has given consent for the access.

**Note:** The several Read account data transactions are own transactions following [XS2A-OR], thus a transaction identification will only be used several times in case of pagination while reading transaction lists/account statements.

### Consent Models

This specification supports three different consent models, cp. also [XS2A-OR]:

- Detailed Consent

  The Consent Management is handled between TPP and PSU. The TPP is submitting then the detailed consent information – PSU identification, services and account numbers affected – to the ASPSP for authorisation by the PSU. The ASPSP is displaying the consent details to the PSU when performing the SCA.

- Global Consent

  The Consent Management is handled between TPP and PSU. The TPP is submitting then a global consent information, which is only the PSU identification, to the ASPSP for authorisation by the PSU. The ASPSP is displaying only the general access to the PSU's account to the PSU when performing the SCA.

- Bank Offered Consent

  The TPP is asking the ASPSP to deal with the Consent Management. The ASPSP might ask the PSU for a detailed consent modelling or just for a global consent on all AIS services. This is authorised by the PSU with an SCA. The detailed consent information can be retrieved by the TPP in a following step by reading the corresponding consent object.

**Account Owner Name Delivery: Potential Impact on Consent Model**

The following rules and requirements for the support of this service apply.

- An ASPSP may deliver the account owner name without any extension to the consent model defined above.

or

- An ASPSP may require an explicit consent extension by the PSU to deliver the account owner name.

If an ASPSP offers the Detailed and the Global Consent Model, then the ASPSP is mandated to offer the extension for both models if it is offered for one of these models.

The offer of the consent extension model for the consent for the available accounts is independent from the above requirement, since it also depends on the fact whether the account owner name is delivered in the payment account overview.

The provision of this service by an ASPSP might depend on the fact that the account owner name is also delivered in online channels of the ASPSP.

## 6.1 Account Information Service Flows

As for the payment initiation, please note that the following flows do not cover all possible variances and are exemplary flows. Especially the flows for

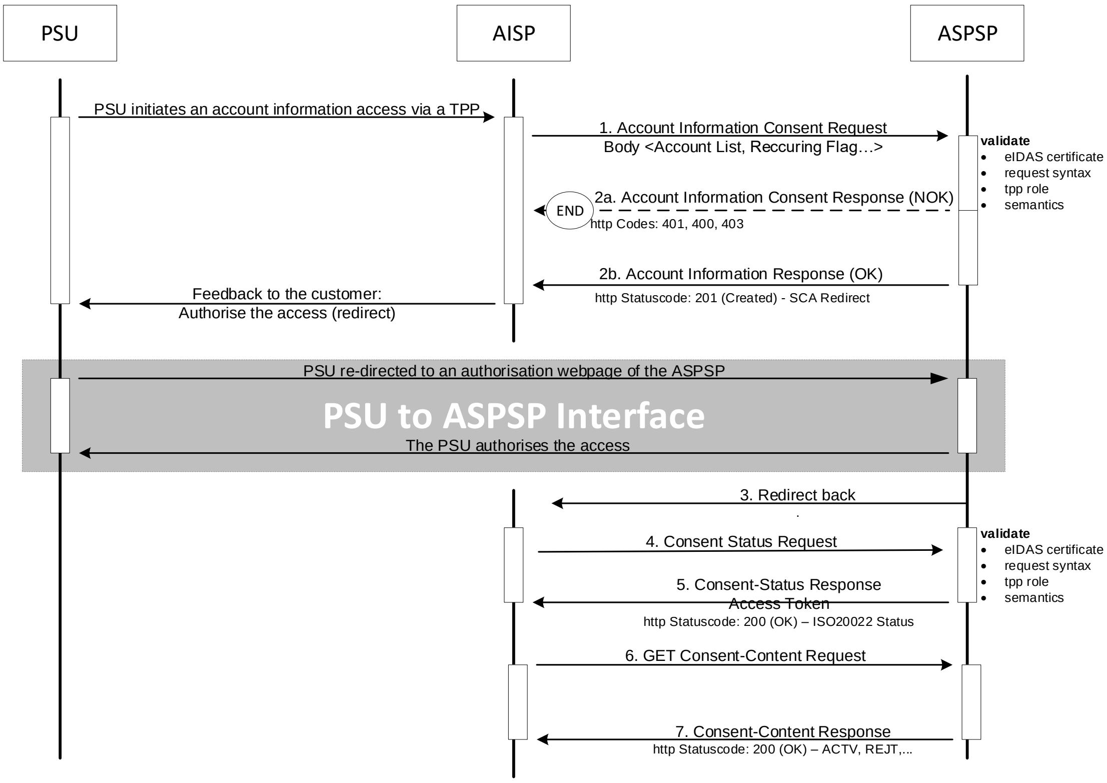
- Redirect and OAuth2 SCA Approach with an explicit start of the Authorisation Process or
- Flows with integrating an explicit confirmation of an authorisation resource

are not shown, since they are following exactly the flow logic as described the Payment Initiation Flows, cp. Section 5.1.

### 6.1.1 Account Information Consent Flow

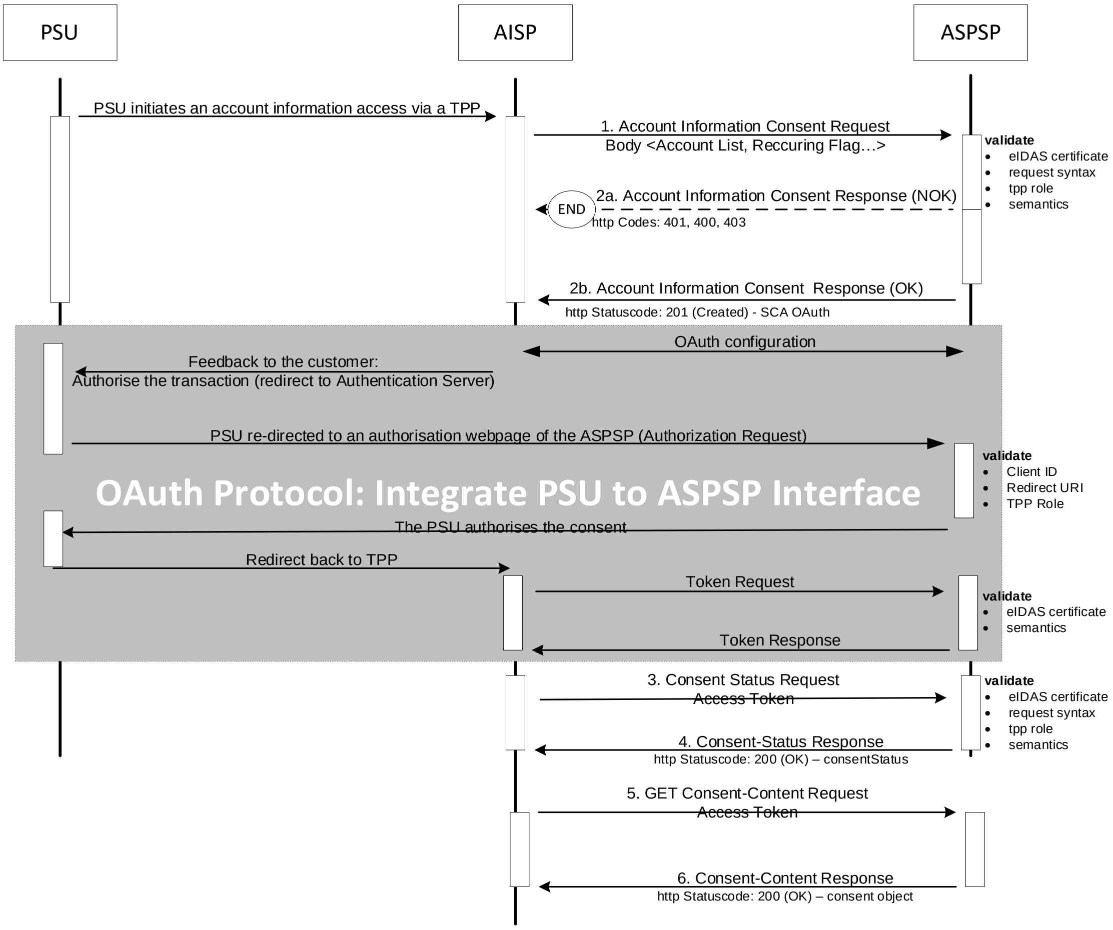
#### 6.1.1.1 Redirect SCA Approach: Implicit Start of the Authorisation Process

If the ASPSP supports the Redirect SCA Approach, the message flow within the Account Information Consent sub-service is simple. The Account Information Consent Request is followed by a redirection to the ASPSP SCA authorisation site. A status or content request on the created consent resource might be requested by the TPP after the session is re-redirected to the TPP's system.


PSU
AISP
ASPSP
PSU initiates an account information access via a TPP
1. Account Information Consent Request
Body <Account List, Reccuring Flag…>
validate
eIDAS certificate
request syntax
tpp role
semantics
END
2a. Account Information Consent Response (NOK)
http Codes: 401, 400, 403
2b. Account Information Response (OK)
http Statuscode: 201 (Created) - SCA Redirect
Feedback to the customer:
Authorise the access (redirect)
PSU re-directed to an authorisation webpage of the ASPSP
PSU to ASPSP Interface
The PSU authorises the access
3. Redirect back
4. Consent Status Request
validate
eIDAS certificate
request syntax
tpp role
semantics
5. Consent-Status Response
Access Token
http Statuscode: 200 (OK) – ISO20022 Status
6. GET Consent-Content Request
7. Consent-Content Response
http Statuscode: 200 (OK) – ACTV, REJT,...

### 6.1.1.2 OAuth2 SCA Approach: Implicit Start of the Authorisation Process

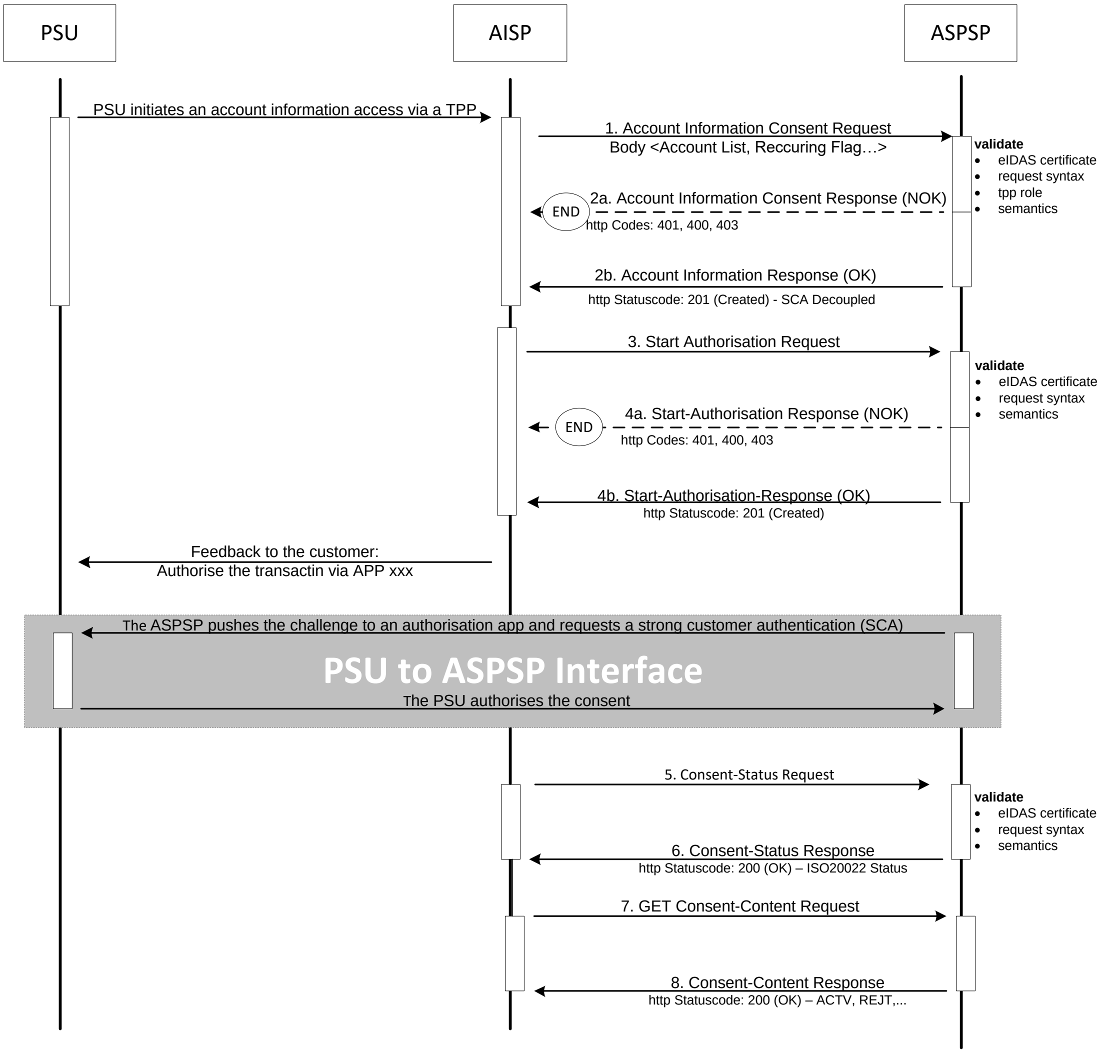
If the ASPSP supports the OAuth2 SCA Approach, the flow is very similar to the Redirect SCA Approach. Instead of redirecting the PSU directly to an authentication server, the OAuth2 protocol is used for the transaction authorisation process. In the following, a flow is shown, where the Authorisation Process in the NextGenPSD2 API has been implicitly started, cp. 5.1.5.


PSU
AISP
ASPSP
PSU initiates an account information access via a TPP
1. Account Information Consent Request
Body <Account List, Reccuring Flag…>
validate
• eIDAS certificate
• request syntax
• tpp role
• semantics
END
2a. Account Information Consent Response (NOK)
http Codes: 401, 400, 403
2b. Account Information Consent Response (OK)
http Statuscode: 201 (Created) - SCA OAuth
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 consent
Redirect back to TPP
Token Request
validate
• eIDAS certificate
• semantics
Token Response
3. Consent Status Request
Access Token
validate
• eIDAS certificate
• request syntax
• tpp role
• semantics
4. Consent-Status Response
http Statuscode: 200 (OK) – consentStatus
5. GET Consent-Content Request
Access Token
6. Consent-Content Response
http Statuscode: 200 (OK) – consent object

### 6.1.1.3 Decoupled SCA Approach: Explicit 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 account access consent e.g. via a dedicated mobile app. 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 account access".

After the SCA between ASPSP and PSU, the TPP then needs to ask for the result of the transaction.


PSU
AISP
ASPSP
PSU initiates an account information access via a TPP
1. Account Information Consent Request
Body <Account List, Reccuring Flag…>
validate
• eIDAS certificate
• request syntax
• tpp role
• semantics
END
2a. Account Information Consent Response (NOK)
http Codes: 401, 400, 403
2b. Account Information Response (OK)
http Statuscode: 201 (Created) - SCA Decoupled
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 transactin 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 consent
5. Consent-Status Request
validate
• eIDAS certificate
• request syntax
• semantics
6. Consent-Status Response
http Statuscode: 200 (OK) – ISO20022 Status
7. GET Consent-Content Request
8. Consent-Content Response
http Statuscode: 200 (OK) – ACTV, REJT,...

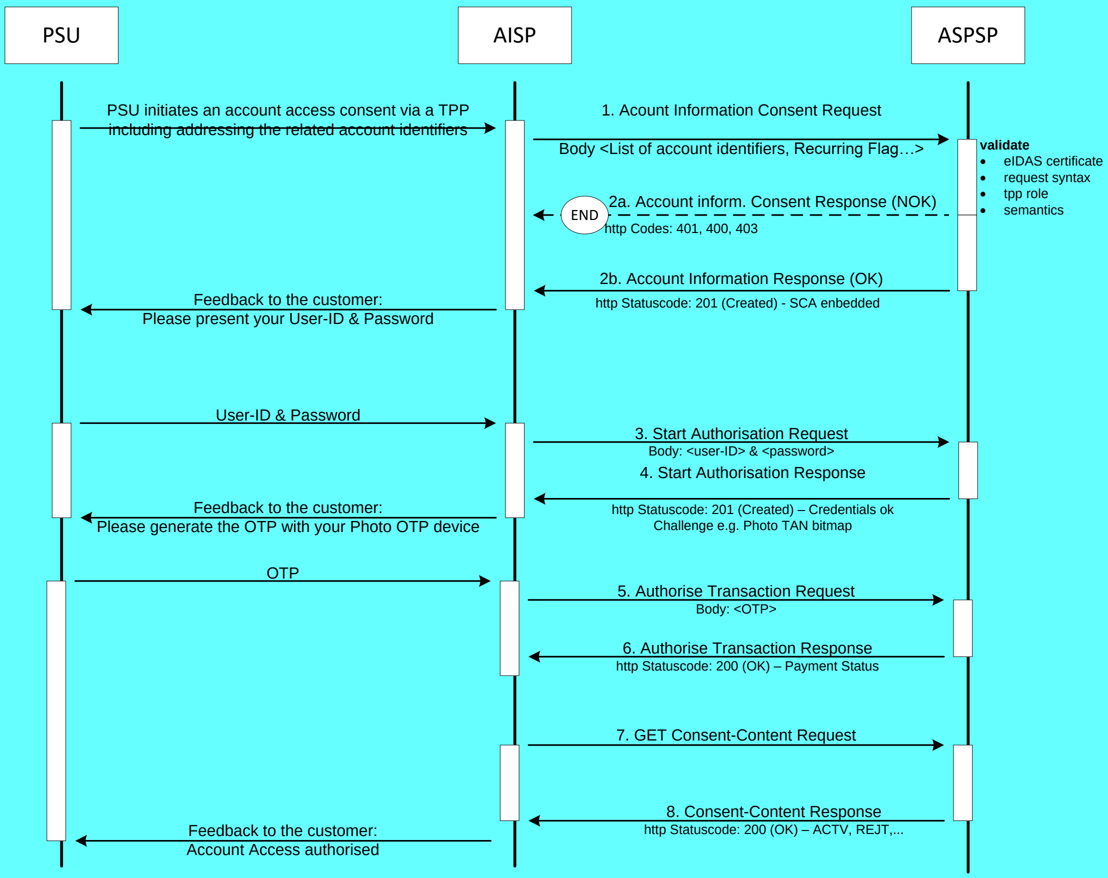
### 6.1.1.4 Embedded SCA Approach with only one SCA method available

In the following, several exemplary flows are shown, where the ASPSP has chosen to process the SCA methods for the consent approval 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.

> **Remark:** In case where OAuth2 is requested by the ASPSP as a pre-step to replace the PSU- and password by an access token, the sequence of the PSU authentication

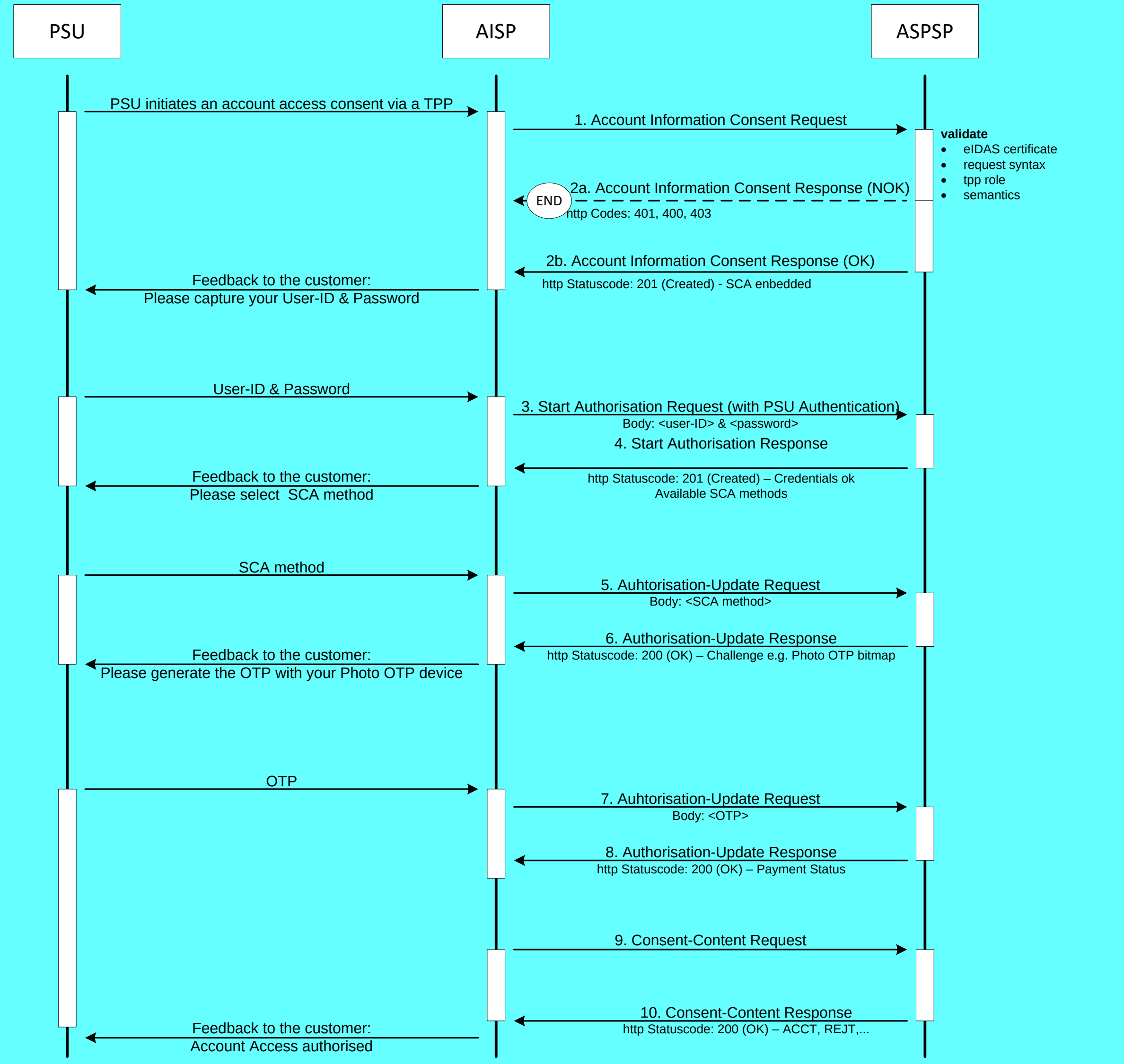
with the first authentication factor is omitted. This applies for all examples for the Embedded SCA Approach.

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
AISP
ASPSP
PSU initiates an account access consent via a TPP including addressing the related account identifiers
1. Acount Information Consent Request
Body <List of account identifiers, Recurring Flag…>
validate
• eIDAS certificate
• request syntax
• tpp role
• semantics
END
2a. Account inform. Consent Response (NOK)
http Codes: 401, 400, 403
2b. Account Information Response (OK)
http Statuscode: 201 (Created) - SCA enbedded
Feedback to the customer:
Please present your User-ID & Password
User-ID & Password
3. Start Authorisation Request
Body: <user-ID> & <password>
4. Start Authorisation Response
http Statuscode: 201 (Created) – Credentials ok
Challenge e.g. Photo TAN bitmap
Feedback to the customer:
Please generate the OTP with your Photo OTP device
OTP
5. Authorise Transaction Request
Body: <OTP>
6. Authorise Transaction Response
http Statuscode: 200 (OK) – Payment Status
7. GET Consent-Content Request
8. Consent-Content Response
http Statuscode: 200 (OK) – ACTV, REJT,...
Feedback to the customer:
Account Access authorised

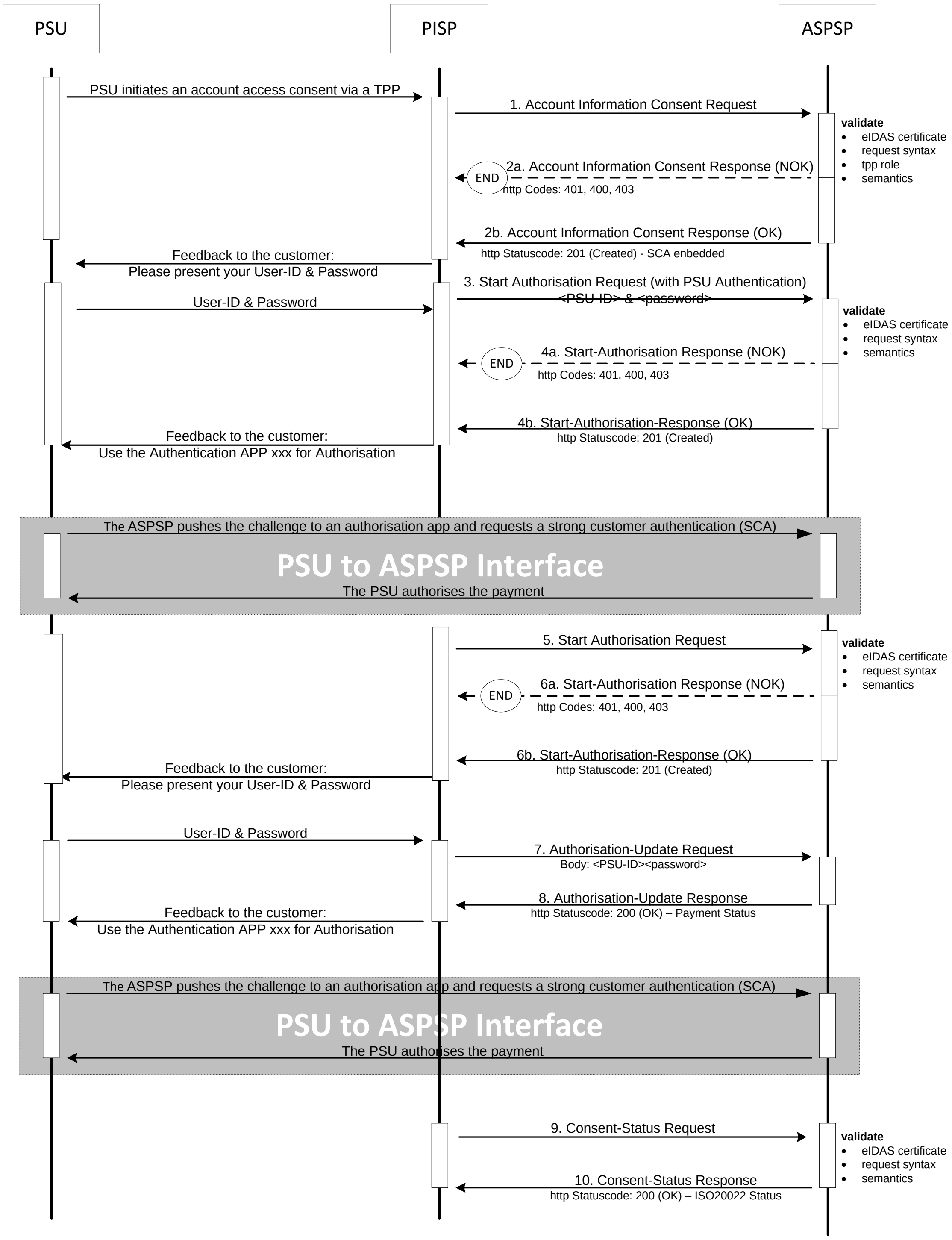
### 6.1.1.5 Embedded SCA Approach with Selection of a SCA method

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
AISP
ASPSP
PSU initiates an account access consent via a TPP
1. Account Information Consent Request
validate
• eIDAS certificate
• request syntax
• tpp role
• semantics
END
2a. Account Information Consent Response (NOK)
http Codes: 401, 400, 403
2b. Account Information Consent Response (OK)
http Statuscode: 201 (Created) - SCA enbedded
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 (Created) – Credentials ok
Available SCA methods
Feedback to the customer:
Please select SCA method
SCA method
5. Auhtorisation-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 the OTP with your Photo OTP device
OTP
7. Auhtorisation-Update Request
Body: <OTP>
8. Authorisation-Update Response
http Statuscode: 200 (OK) – Payment Status
9. Consent-Content Request
10. Consent-Content Response
http Statuscode: 200 (OK) – ACCT, REJT,...
Feedback to the customer:
Account Access authorised

### 6.1.1.6 Multilevel SCA Approach: Example Decoupled SCA Approach

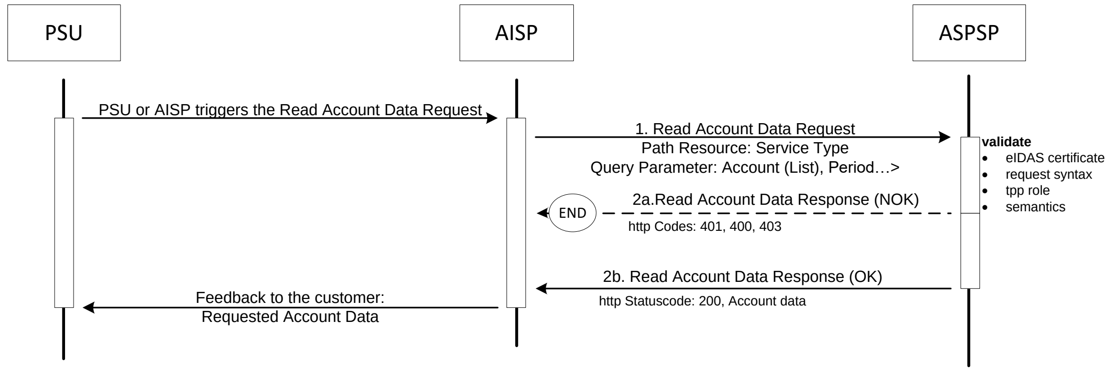
The multilevel SCA Approach flows for the Establish Consent Requests will follow exactly the same pattern as for the Payment Initiation, cp. Section 5.1.12. Whereas the Redirect SCA Approach was used there as an example, the following flow will give an example for the Decoupled SCA Approach:


PSU
PISP
ASPSP
PSU initiates an account access consent via a TPP
1. Account Information Consent Request
validate
• eIDAS certificate
• request syntax
• tpp role
• semantics
END
2a. Account Information Consent Response (NOK)
http Codes: 401, 400, 403
2b. Account Information Consent Response (OK)
http Statuscode: 201 (Created) - SCA enbedded
Feedback to the customer:
Please present your User-ID & Password
3. Start Authorisation Request (with PSU Authentication)
<PSU-ID> & <password>
User-ID & Password
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:
Use the Authentication APP xxx for Authorisation
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
5. Start Authorisation Request
validate
• eIDAS certificate
• request syntax
• semantics
END
6a. Start-Authorisation Response (NOK)
http Codes: 401, 400, 403
6b. Start-Authorisation-Response (OK)
http Statuscode: 201 (Created)
Feedback to the customer:
Please present your User-ID & Password
User-ID & Password
7. Authorisation-Update Request
Body: <PSU-ID><password>
8. Authorisation-Update Response
http Statuscode: 200 (OK) – Payment Status
Feedback to the customer:
Use the Authentication APP xxx for Authorisation
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
9. Consent-Status Request
validate
• eIDAS certificate
• request syntax
• semantics
10. Consent-Status Response
http Statuscode: 200 (OK) – ISO20022 Status

Note, that in this example the ASPSP asks in Step 6b the TPP to add PSU-ID and password, since it was not uploaded together with the Start Authorisation Process.

### 6.1.2 Read Account Data Flow

The Read Account Data flow is independent from the corresponding Consent Management flow. It is a simple Request/Response process as follows:


PSU
AISP
ASPSP
PSU or AISP triggers the Read Account Data Request
1. Read Account Data Request
Path Resource: Service Type
Query Parameter: Account (List), Period…>
validate
• eIDAS certificate
• request syntax
• tpp role
• semantics
2a.Read Account Data Response (NOK)
END
http Codes: 401, 400, 403
2b. Read Account Data Response (OK)
http Statuscode: 200, Account data
Feedback to the customer:
Requested Account Data

## 6.2 Data Overview Account Information Service

The following table defines the technical description of the abstract data model as defined [XS2A OR] for the account information 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, resp. are taken from eIDAS certificates.
- The "Usage" column gives an overview on the usage of data elements in the different API Calls. Within [XS2A-OR], the XS2A calls are described as abstract API calls. These calls will be technically realised as HTTP POST, PUT, DELETE and GET commands. The calls are divided into the following calls:
    - Establish Consent Request, which shall be the first API Call for every transaction within XS2A Account Information service.
    - 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 of a second factor is needed.
    - The Read Data Request is the request to retrieve Account Information data, which is addressed to different endpoints with different parameters.
    - The Status Request is used 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 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).

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

| Data element | Attribute encoding | Location | | | | | Usage | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Path | Query Param. | Header | Body | Certificate | Establ.. Cons.Req. | Establ. Cons. Resp. | Upd. Data Req. | Upd. Data Resp | Auth. Req. | Auth Resp. | Status Req. | Status Resp. | Read Data Req. | Read Data Resp |
| Provider Identification | | x | | | | | m | | m | | m | | m | | m | |
| TPP Registration Number | | | | | | x | m | | m | | m | | m | | m | |
| TPP Name | | | | | | x | m | | m | | m | | m | | m | |
| TPP Role | | | | | | x | m | | m | | m | | m | | m | |
| TPP National Competent Authority | | | | | | x | m | | m | | m | | m | | m | |
| Request Identification | X-Request-ID | | | x | | | m | m | m | m | m | m | m | m | m | m |
| Resource ID | consentId | | | | x | | | m | | | | | | | | |
| Resource ID[7] | | x | | | | | | | m | | m | | m | | | |
| Resource-ID[8] | Consent-ID | | | x | | | | | | | | | | | m | |
| Access Token (from optional OAuth2) | Authorization | | | x | | | c | | c | | c | | c | | c | |

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

[8] Please note that the consent identification is addressed by different syntax depending of where it is transported.

| Data element | Attribute encoding | Location | | | | | Usage | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Path | Query Param. | Header | Body | Certificate | Establ.. Cons.Req. | Establ. Cons. Resp. | Upd. Data Req. | Upd. Data Resp | Auth. Req. | Auth Resp. | Status Req. | Status Resp. | Read Data Req. | Read Data Resp |
| TPP Signing Certificate Data | TPP-Signature-Certificate | | | x | | | c | | c | | c | | c | | c | |
| TPP Signing Electronic Signature | Signature | | | x | | | c | | c | | c | | c | | c | |
| Further signature related data | Digest | | | x | | | c | | c | | c | | c | | c | |
| ASPSP-SCA-Approach | ASPSP-SCA-Approach | | | x | | | | c | | c | | | | | | |
| Transaction Status | consentStatus | | | | x | | | m | | m | | m | | m | | |
| SCA Status | scaStatus | | | | x | | | | | | | | | o | | |
| PSU Message Information | psuMessage | | | | x | | | o | | o | | o | | o | | o |
| TPP Message Information | tppMessages | | | | x | | | o | | 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 Type | PSU-Corporate-ID-Type | | | | | | c | | c | | c | | c | | | |
| PSU Password | psuData.password | | | | x | | | | c | | | | | | | |
| Available SCA Methods | scaMethods | | | | x | | | c | | c | | | | | | |
| Chosen SCA Method | chosenScaMethod | | | | x | | | | c | | | | | | | |
| PSU Authentication Data | psuData.authentication | | | | x | | | | | | m | | | | | |
| SCA Challenge Data | challengeData | | | | x | | | c | | c | | | | | | |

| **Data element** | **Attribute encoding** | **Location** | | | | | **Usage** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Path | Query Param. | Header | Body | Certificate | Establ.. Cons.Req. | Establ. Cons. Resp. | Upd. Data Req. | Upd. Data Resp | Auth. Req. | Auth Resp. | Status Req. | Status Resp. | Read Data Req. | Read Data Resp |
| IP Address PSU | PSU-IP-Address | | | x | | | m | | o | | o | | o | | c | |
| PSU IP Port | PSU-IP-Port | | | x | | | o | | o | | o | | o | | o | |
| Further PSU related Information | PSU-Accept | | | x | | | o | | o | | o | | o | | o | |
| | PSU-Accept-Charset | | | x | | | o | | o | | o | | o | | o | |
| | PSU-Accept-Encoding | | | x | | | o | | o | | o | | o | | o | |
| | PSU-Accept-Language | | | x | | | o | | o | | o | | o | | o | |
| | PSU-Http-Method | | | x | | | o | | o | | o | | o | | o | |
| | PSU-Device-ID | | | x | | | o | | o | | o | | o | | o | |
| PSU User Agent | PSU-User-Agent | | | x | | | o | | o | | o | | o | | o | |
| GEO Information | PSU-Geo-Location | | | x | | | o | | o | | o | | o | | o | |
| Redirect URL ASPSP | _links.scaRedirect | | | | x | | | c | | | | | | | | |
| Redirect Preference | TPP-Redirect-Preferred | | | x | | | o | | | | | | | | | |
| Decoupled Preference | TPP-Decoupled-Preferred | | | x | | | o | | | | | | | | | |
| Redirect URL TPP | TPP-Redirect-URI | | | x | | | c | | | | | | | | | |
| Authorisation Preference | TPP-Explicit-Authorisation-Preferred | | | x | | | o | | | | | | | | | |
| TPP Notification URI | TPP-Notification-URI | | | x | | | o | | | | | | | | | |
| TPP Notification Content Preference | TPP-Notification-Content-Preferred | | | x | | | o | | | | | | | | | |

| Data element | Attribute encoding | Location | | | | | Usage | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Path | Query Param. | Header | Body | Certificate | Establ.. Cons.Req. | Establ. Cons. Resp. | Upd. Data Req. | Upd. Data Resp | Auth. Req. | Auth Resp. | Status Req. | Status Resp. | Read Data Req. | Read Data Resp |
| TPP Brand Information | TPP-Brand-Logging-Information | | | x | | | o | | | | | | | | | |
| PSU Account | account | | | | x | | | | | | | | | | c | |
| PSU Account List | access | | | | x | | m | | | | | | | | | |
| Date From | dateFrom | | x | | | | | | | | | | | | c | |
| Date To | dateTo | | x | | | | | | | | | | | | c | |
| Transaction From | entryReferenceFrom | | x | | | | | | | | | | | | o | |
| Booking Status | bookingStatus | | x | | | | | | | | | | | | o | |
| Delta Indicator | deltaList | | x | | | | | | | | | | | | o | |
| With Balance Flag | withBalance | | x | | | | | | | | | | | | o | |
| Validity Period | validUntil | | | | x | | m | | | | | | | | | |
| Frequency | frequencyPerDay | | | | x | | m | | | | | | | | | |
| Recurring Indicator | recurringIndicator | | | | x | | m | | | | | | | | | |
| Combined service | combinedService Indicator | | | | x | | m | | | | | | | | | |

**Remark:** The upper table refers to the "Acccount Information Consent Request" referring dedicated accounts, cp. Section 6.3.1.1.

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

**Remark:** The AIS and PIS services are 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.

### 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.

### Multi-currency Account Specifics for Account Information

The methods on multicurrency accounts for account information differ in the inter-face due to the fact, that a collection of accounts is addressed. In the following the differences are described on abstract level.

### Multicurrency Accounts in Submission of Consents

Multicurrency accounts are addressed by just using the external account identifier in the submission of a consent on dedicated accounts, without specifying a currency. Asking for the consent to retrieve account information data of a multicurrency accounts implies getting it for all sub-accounts.

### Multicurrency Accounts in Reading Accounts or Account Details

The ASPSP will decide in its implementation whether to grant data access to a multicurrency account on aggregation level, on aggregation and sub-account level, or only on sub-account level.

### Multicurrency Accounts in Reading Balances

The consequence for this function is that an array of balances of all sub-accounts are returned, if a multicurrency account is addressed on aggregation level. The currency of the respective sub-account is implicitly provided as the currency of the balanceAmount element within the balance.

### Multicurrency Accounts in Reading Transactions

The consequence for this function is that the list of transactions will contain all transactions of all sub-accounts, if a multicurrency account is addressed on aggregation level. In this case the payment transactions contained in the report may have different transaction currencies.

## 6.3 Establish Account Information Consent

In this section, the Establish Account Information Consent process is defined for the XS2A Interface.

### 6.3.1 Account Information Consent Request

#### 6.3.1.1 Consent Request on Dedicated Accounts

**Call**

```
POST /v1/consents
```

Creates an account information consent resource at the ASPSP regarding access to accounts specified in this request.

**Side Effects**

When this Consent Request is a request where the "recurringIndicator" equals true, and if it exists already a former consent for recurring access on account information for the addressed PSU and potentially addressed corporate identification submitted by this TPP, then the former consent automatically expires as soon as the new consent request is authorised by the PSU.

Please note that in the context of multilevel SCA, the side effect may only apply, if a PSU with a given PSU-ID is submitting a new recurring consent and a second recurring consent exists already which had been submitted under the same PSU and potentially corporate identification.

There are no expiration side effects foreseen for Consent Requests where the "recurringIndicator" equals false.

**Query Parameters**

No specific query parameter.

**Request Header**

| Attribute | Type | Condition | Description |
|---|---|---|---|
| X-Request-ID | UUID | Mandatory | ID of the request, unique to the call, as determined by the initiating party. |
| PSU-ID | String | Conditional | Client ID of the PSU in the ASPSP client interface. Might be mandated in the ASPSP's documentation.<br>It might be contained even if an OAuth2 based authentication was performed in a pre-step In this case the ASPSP might check whether PSU-ID and token match, according to ASPSP documentation." |

| Attribute | Type | Condition | Description |
|---|---|---|---|
| PSU-ID-Type | String | Conditional | Type of the PSU-ID, needed in scenarios where PSUs have several PSU-IDs as access possibility. |
| PSU-Corporate-ID | String | Conditional | Might be mandated in the ASPSP's documentation. Only used in a corporate context. |
| PSU-Corporate-ID-Type | String | Conditional | Might be mandated in the ASPSPs documentation. Only used in a corporate context. |
| PSU-IP-Address | String | Mandatory | The forwarded IP Address header field consists of the corresponding HTTP request IP Address field between PSU and TPP.<br>If not available, the TPP shall use the IP Address used by the TPP when submitting this request. |
| Authorization | String | Conditional | If OAuth2 has been chosen as pre-step to authenticate the PSU. |
| TPP-Redirect-Preferred | Boolean | Optional | If it equals "true", the TPP prefers a redirect over an embedded SCA approach.<br>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 choice of the SCA procedure by the TPP/PSU.<br>If the parameter is not used, the ASPSP will choose the SCA approach to be applied depending on the parameter TPP-Decoupled-Preferred and the SCA method chosen by the TPP/PSU. |
| TPP-Decoupled-Preferred | Boolean | Optional | If it equals "true", the TPP prefers a decoupled SCA approach.<br>If it equals "false", the TPP prefers not to use the decoupled approach for SCA. The |