8 Signing Baskets

8.1 Establish Signing Basket Request

`POST /v1/signing-baskets/`

Generates a signing basket

Path Parameters

None.

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.
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 an 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 is 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.

Attribute	Type	Condition	Description
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 an preceding AIS service in the same session.
Consent-ID	String	Optional	This data element may be contained, if the signing basket transaction is part of a session, i.e. combined AIS/PIS service. This then contains the "consentId" of the related AIS one off consent, which was performed prior to this bulk signing.
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 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-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. If it equals "false", the TPP prefers not to use the decoupled approach for SCA. The ASPSP will

Attribute	Type	Condition	Description
			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. 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	Must equal "true", if contained. Remark: No optimisation processes for creating authorisation resources for signing baskets

Attribute	Type	Condition	Description
			implicitly, since anyhow several calls have been submitted.
TPP-Notification-URI	String	Optional	URI for the Endpoint of the TPP-API to which the status of the basket 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.

Request Body

Attribute	Type	Condition	Description
paymentIds	Array of String	Optional	A non empty array of paymentIds

Attribute	Type	Condition	Description
consentIds	Array of String	Optional	A non empty array of consentIds

The body shall contain at least one entry.

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 non payment related values defined in Section 14.25 might be used like RCVD or ACTC. For a list of all transactionStatus codes permitted for signing baskets, cp. Section 8.3.

Attribute	Type	Condition	Description
basketId	String	Mandatory	resource identification of the generated signing basket 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 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 "startAuthorisationWith PsuAuthentication" 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.

Attribute	Type	Condition	Description
			Type of links admitted in this response, (further links might be added for ASPSP defined extensions): "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. "startAuthorisationWithEncrypted PsuAuthentication": The link to the authorisation end-point, where the authorisation sub-resource has to be generated while uploading the encrypted PSU authentication data. "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" :

Attribute	Type	Condition	Description
			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.
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/signing-baskets
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

{
   "paymentIds": ["123qwert456789", "12345qwert7899"]
}
```

Response (always with explicit authorisation start)

```
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/signing-
baskets/1234-basket-567
```

```
Content-Type:           application/json

{
  "transactionStatus": "RCVD",
  "basketId": "1234-basket-567",
  "_links": {
        "self": {"href": "/psd2/v1/signing-baskets/1234-basket-567"},
        "status": {"href": "/psd2/v1/signing-baskets/1234-basket-
567/status"},
        "startAuthorisation": {"href": "/psd2/v1/signing-baskets/1234-
basket-567/authorisations"}
  }
}
```

8.2 Get Signing Basket Request

Call

```
GET /v1/signing-baskets/{basketId}
```

Returns the content of a signing basket object.

Path Parameters

Attribute	Type	Description
basketId	String	ID of the corresponding signing basket object.

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.
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

Attribute	Type	Condition	Description
			current PIS transaction or in a preceding AIS service in the same session, if no such OAuth2 SCA approach was chosen in the current signing basket 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
payments	array of paymentId	Optional	payment initiations which shall be authorised through this signing basket.
consents	array of consentId	Optional	consent objects which shall be authorised through this signing basket.
transactionStatus	Transaction Status	Mandatory	Only the not explicitly payment related codes like RCVD, PATC, ACTC, RJCT are used. For a list of all transactionStatus codes permitted for signing baskets, cp. Section 8.3.
_links	Links	Optional	The ASPSP might integrate hyperlinks to indicate next (authorisation) steps to be taken.

Example

Request

```
GET https://api.testbank.com/psd2/v1/signing-baskets/1234-basket-567
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                   Sun, 06 Aug 2017 15:05:46 GMT
```

Response

```
HTTP/1.x 200 Ok
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                   Sun, 06 Aug 2017 15:05:47 GMT
Content-Type:           application/json

{
"payments": ["1234pay567","1234pay568","1234pay888"],
"transactionStatus": "ACTC"
}
```

8.3 Get Signing Basket Status Request

Call

```
GET /v1/signing-baskets/{basketId}/status
```

Returns the status of a signing basket object.

Path Parameters

Attribute	Type	Description
basketId	String	ID of the corresponding signing basket object.

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.
Authorization	String	Conditional	Is contained only, if an OAuth2 based authentication was performed in a pre-step or an

Attribute	Type	Condition	Description
			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 signing basket 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
transactionStatus	Transaction Status	Mandatory	Only the codes RCVD, PATC, ACTC, CANC and RJCT are supported for signing baskets.

Example

Request

```
GET https://api.testbank.com/psd2/v1/signing-baskets/1234-basket-567/status
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                   Sun, 06 Aug 2017 15:05:49 GMT
```

Response

```
HTTP/1.x 200 Ok
```

```
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                   Sun, 06 Aug 2017 15:05:51 GMT
Content-Type:           application/json

{
"transactionStatus": "ACTC"
}
```

8.4 Multi-level SCA for Signing Baskets

The Establish Signing Basket Request defined above is independent from the need of one or multilevel SCA processing, i.e. independent from the number of authorisations needed for the execution of all transactions contained in the basket. In contrast, the Establish Signing Basket Response defined above in this section are specific to the processing of one SCA. processing. In the following the background is explained on diverging requirements on the Establish Signing Basket Response message.

If any data is needed for starting the next action, like selecting an SCA method, this action is not supported through a hyperlink 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 in case of Multi-Level SCA needed

Attribute	Type	Condition	Description
transactionStatus	Transaction Status	Mandatory	The non payment related values defined in Section 14.25 might be used like RCVD, ACTC, PATC, CANC or RJCT
basketId	String	Mandatory	resource identification of the generated signing basket resource.
_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):

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

8.5 Cancellation of Signing Baskets

A cancellation of a Signing Basket is only permitted where no (partial) authorisation has been applied for the Signing Basket.

Call

```
DELETE /v1/signing-baskets/{basketId}
```

Deletes a created signing basket if it is not yet (partially) authorised.

Path Parameters

Attribute	Type	Description
basketId	String	Contains the resource-ID of the signing basket to be deleted.

Query Parameters

No specific query parameters.

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 SCA has been used in a pre-step.

Request Body

No Request Body.

Response Code

The HTTP response code is 204 in case of successful deletion.

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

No Response Body

Example

Request

```
DELETE https://api.testbank.com/psd2/v1/signing-baskets/qwer3456tzui9876
X-Request-ID            99391c7e-ad88-49ec-a2ad-99ddcb1f7757
Date                    Sun, 13 Aug 2017 17:05:37 GMT
```

Response

```
HTTP/1.x 204 No Content
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7757
Date:                   Sun, 13 Aug 2017 17:05:38 GMT
```

9 Sessions: Combination of AIS and PIS Services

The implementation of sessions in the sense of [XS2A-OR], i.e. the combination of AIS and PIS services is an optional feature of this interface. The ASPSP will inform about the support by its PSD2 documentation.

This feature might be relevant where account information services are needed within a payment initiation, especially for batch booking banks. In this case, a consent to access the corresponding account information is needed, cp. Section 6.3. The corresponding GET method to read the account data is using there the header parameter "Consent-ID". The TPP then can use this Consent-ID parameter also in the POST method when applying the Payment Initiation Request, cp. Section 5.3. A pre-requisite to use the "Consent-ID" in the subsequent Payment Initiation Request is that the flag "combinedServiceIndicator" in the Account Information Consent Request was set, cp. Section 6.3.1.

The usage of the "Consent-ID" in the subsequent Payment Initiation Request will then yield to not again ask for a first authentication factor, so the ASPSP will not again provide the PSU authentication link. In a case of SCA exemption for the corresponding payment, this can yield to a situation where no further PSU authentication is needed – the payment will then be executed without further confirmation.

In a context, where the consent management for account access is fully provided by the OAuth2 model, the corresponding access tokens will support this feature analogously.

10 Confirmation of Funds Service

10.1 Overview Confirmation of Funds Service

The following table defines the technical description of the abstract data model as defined [XS2A-OR] for the three PSD2 services. 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 services and API Calls. Within [XS2A-OR], the XS2A calls are described as abstract API calls. These calls will be technically realised as HTTP POST command. The calls are divided into the following calls:
    - Confirmation Request, which is the only API Call for every transaction within the Confirmation of Funds service.

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).

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

Data element	Attribute encoding	Location				Usage	
		Path	Header	Body	Certificate	Conf. Req.	Conf Resp.
Provider Identification		x				m	
TPP Registration Number					x	m	
TPP Name					x	m	
TPP Role					x	m	
TPP National Competent Authority					x	m	
Request Identification	X-Request-ID		x			m	m
Consent ID	Consent-ID		x			c	
TPP Certificate Data	TPP-Signature-Certificate		x			c	
Further signature related data	Digest		x			c	
TPP Electronic Signature	Signature		x			c	
TPP Message Information	tppMessages			x			o
Card Number	cardNumber			x		o	
Account Number	account			x		m	
Name Payee	payee			x		o	
Transaction Amount	instructedAmount			x		m	

10.2 Confirmation of Funds Request

Call

```
POST /v1/funds-confirmations
```

Creates a confirmation of funds request at the ASPSP.

Query Parameter

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.
Authorization	String	Optional	This field might be used in case where a consent was agreed between ASPSP and PSU through an OAuth2 based protocol, facilitated by the TPP.
Consent-ID	String	Conditional	Shall be provided if the consent of the PSU has been provided through the consent process as defined in [XS2A-COFC]. Otherwise not used.
Digest	cp. Section 12	Conditional	Is contained if and only if the "Signature" element is contained in the header of the request.
Signature	cp Section 12	Conditional	A signature of the request by the TPP on application level. This might be mandated by ASPSP.
TPP-Signature-Certificate	String	Conditional	The certificate used for signing the request, In base64 encoding.

Request Body

Attribute	Type	Condition	Description
cardNumber	Max35Text	Optional	Card Number of the card issued by the PIISP. Should be delivered if available.
account	Account Reference	Mandatory	PSU's account number.
payee	Max70Text	Optional	The merchant where the card is accepted as an information to the PSU.
instructedAmount	Amount	Mandatory	Transaction amount to be checked within the funds check mechanism.

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
fundsAvailable	Boolean	Mandatory	Equals true if sufficient funds are available at the time of the request, false otherwise.

The following rules will apply in interpreting the Confirmation of Funds Request for multicurrency accounts:

The additional card number might support the choice of the sub-account.

If no card number, but the PSU account identifier is contained: check on default account registered by customer.

If no card number but the PSU and the account identifier with currency is contained: check the availability of funds on the corresponding sub-account.

If card number and the PSU account identifier is contained: check on sub-account addressed by card, if the addressed card is registered with one of the sub-accounts.

If the card number is not registered for any of the sub-accounts, or if the card number is registered for a different sub-account the card number might be ignored.

Example

```
POST https://api.testbank.com/psd2/v1/funds-confirmations
Content-Type:           application/json
X-Request-ID:           99391c7e-ad88-49ec-a2ad-99ddcb1f7721
Date:                   Sun, 06 Aug 2017 15:02:37 GMT

{  "cardNumber": "12345678901234",
   "account": {"iban": "DE23100120020123456789"},
```

```
   "instructedAmount": {"currency": "EUR", "amount": "123"}
}
```

Response Body

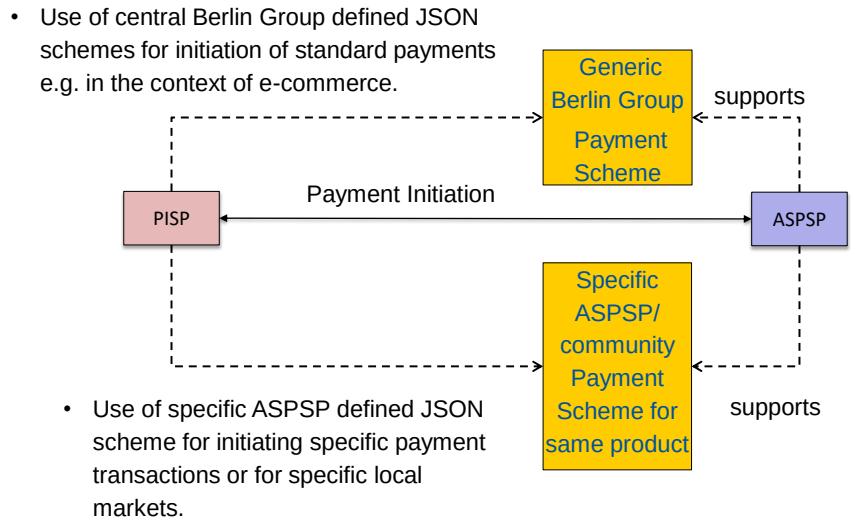
```
{"fundsAvailable": true}
```

11 Core Payment Structures

For core payment products in the European market, this document is defining JSON structures, which will be supported by all ASPSPs

- offering the corresponding payment products to their customers and
- providing JSON based payment endpoints, cp Sections 5.3.1 and 5.3.3.1.

At the same time, the ASPSP may offer in addition more extensive JSON structures for the same payment products since they might offer these extensions also in their online banking system.



11.1 Single Payments

The following table first gives an overview on the generic Berlin Group defined JSON structures of standard SEPA payment products for single payments.

Data Element	Type	SCT EU Core	SCT INST EU Core	Target2 Paym. Core	Cross Border CT Core
endToEnd Identification	Max35Text	optional	optional	optional	n.a.
instructionIdentification	Max35Text	n.a.	n.a.	n.a.	n.a.
debtorName	Max70Text	n.a.	n.a.	n.a.	n.a.
debtorAccount (incl. type)	Account Reference	mandatory[11]	mandatory[11]	mandatory[11]	mandatory[11]
debtorId	Max35Text	n.a.	n.a.	n.a.	n.a.
ultimateDebtor	Max70Text	n.a.	n.a.	n.a.	n.a.
instructedAmount (inc. Curr.)	Amount	mandatory	mandatory	mandatory	mandatory
currencyOfTransfer[12]	Currency Code	n.a.	n.a.	n.a.	n.a.
exchangeRateInformation	Payment Exchange Rate	n.a.	n.a.	n.a.	n.a.
creditorAccount	Account Reference	mandatory	mandatory	mandatory	mandatory
creditorAgent	BICFI	optional	optional	optional	conditional[13]
creditorAgentName	Max140Text	n.a.	n.a.	n.a.	n.a.
creditorName	Max70Text	mandatory	mandatory	mandatory	mandatory
creditorId	Max35Text	n.a.	n.a.	n.a.	n.a.
creditorAddress	Address	optional	optional	optional	conditional[14]
creditorNameAnd Address	Max140Text	n.a.	n.a.	n.a.	n.a.
ultimateCreditor	Max70Text	n.a.	n.a.	n.a.	n.a.
purposeCode	Purpose Code	n.a.	n.a.	n.a.	n.a.

[11] ASPSPs might change the condition on the debtor account for SEPA payments to optional as one way to fulfil the requirement according to item 36 of the EBA Opinion of June 2020. This applies to single payments as well as to related recurring payments.

[12] This is a data element to indicate a diverging interbank transaction currency.

[13] This field might be mandated by ASPSPs generally or depending of the creditor's address' country.

[14] This field might be mandated by ASPSPs generally or depending of the creditor's address' country.

Data Element	Type	SCT EU Core	SCT INST EU Core	Target2 Paym. Core	Cross Border CT Core
chargeBearer	Charge Bearer	n.a.	n.a.	optional	conditional[15]
serviceLevel	Service Level Code	n.a.	n.a.	n.a.	n.a.
remittance Information Unstructured	Max140Text	optional	optional	optional	optional
remittance Information Unstructured Array	Array of Max140Text	n.a.	n.a.	n.a.	n.a.
remittance Information Structured	Remittance	n.a.	n.a.	n.a.	n.a.
remittance Information Structured Array	Array of Remittance	n.a.	n.a.	n.a.	n.a.
additionalRemittance Information	Max140Text	n.a.	n.a.	n.a.	n.a.
requestedExecution Date	ISODate	n.a.	n.a.	n.a.	n.a.
requestedExecution Time	ISODateTime	n.a.	n.a.	n.a.	n.a.

The data elements marked with "n.a." are not used in the addressed core services, shared by all ASPSP offering these product, but they can be used in ASPSP or community wide extensions. Extensions of these tables are permitted by this specification

- if they are less restrictive (e.g. set the debtor account to optional) or
- if they open up for more data elements (e.g. open up the structured remittance information, or ultimate data fields.)

  Remark: The debtor account is a mandatory field for a single payment. If bulk payments are use, the debtor account is only used in the introductory part of the bulk structure, cp. Section 11.3.

  Remark: The ASPSP may reject a payment initiation request where additional data elements are used which are not specified.

[15] This field might be mandated by ASPSPs generally or depending of default usage definitions of the ASPSP.

Remark: An example for the above introduced extensions for the SEPA payments are the extensions for the Austrian market as described in [XS2A-DP].

11.2 Future Dated Payments

One example of an extension of the above defined JSON structure is the requested execution date e.g. for SEPA Credit Transfers. This field is n.a. since not all banks or banking communities might support this as a PSD2 core service.

The ASPSP will indicate the acceptance of future dated payments by issuing an ASPSP specific or community specific JSON scheme, where the attribute "requestedExecutionDate" is an optional field.

11.3 Bulk Payments

This specification offers the bulk payment function in JSON encoding as optional endpoint. The format of the bulk payment is an array of single payments, as offered by the ASPSP, preceded by generic payment information applicable to all individual payments contained.

Data Element	Type	Condition	Description
batchBookingPreferred	Boolean	optional	If this element equals true, the PSU prefers only one booking entry. If this element equals false, the PSU prefers individual booking of all contained individual transactions. The ASPSP will follow this preference according to contracts agreed on with the PSU.
debtorAccount (incl. type)	Account Reference	mandatory[16]	
categoryPurposeCode	Category Purpose Code	n.a.	Category Purpose
paymentInformationId	Max35Text	n.a.	Unique identification as assigned by the sending party to unambiguously identify this bulk payment. This attribute may be used by ASPSPs or communities as an optional field. **Remark for Future**: This attribute might be made mandatory in the next version of the specification.
requestedExecutionDate	ISODate	optional	If contained, the payments contained in this bulk will be executed at the addressed date. This field may not be

[16] ASPSPs might change the condition on the debtor account for SEPA bulk payments to optional as one way to fulfil the requirement according to item 36 of the EBA Opinion of June 2020, if this is needed from a compliance point of view. This implementation variant is not recommended due to the complex authorisation processes.

Data Element	Type	Condition	Description
			used together with the field requestedExecutionTime.
requestedExecutionTime	ISODateTime	optional	If contained, the payments contained in this bulk will be executed at the addressed Date/Time. This field may not be used together with the field requestedExecutionDate.
payments	Bulk Entry	mandatory	The Bulk Entry Type is a type which follows the JSON formats for the supported products for single payments, see Section 11.1, excluding the data elements • debtorAccount, • requestedExecutionDate, • requestedExecutionTime. These three data elements may not be contained in any bulk entry.

Example

```
{"batchBookingPreferred": true,
 "debtorAccount": {"iban": "DE40100100103307118608"},
 "requestedExecutionDate": "2018-08-01",
 "payments":
[{JSON based payment initiation 1}, {JSON based payment initiation 2}]}
```

12 Signatures

When an ASPSP requires the TPP to send a digital signature as defined in [signHTTP], chapter 4 in his HTTP-Requests, the signature must obey the following requirements according or additional to [signHTTP], chapter 4.

12.1 "Digest" Header mandatory

When a TPP includes a signature as defined in [signHTTP], chapter 4, he also must include a "Digest" header as defined in [RFC3230]. The "Digest" Header contains a Hash of the message body, if the message does not contain a body, the "Digest" header must contain the hash of an empty bytelist. The only hash algorithms that may be used to calculate the Digest within the context of this specification are SHA-256 and SHA-512 as defined in [RFC5843].

Remark: In case of a multipart message the same method is used to calculate the digest. I.e. a hash of the (whole) message body is calculated including all parts of the multipart message as well as the separators.

12.2 Requirements on the "Signature" Header

As defined in [signHTTP], chapter 4, a "Signature" header must be present. The structure of a "Signature" header is defined in [signHTTP], chapter 4.1, the following table lists the requirements on the "Signature" header from [signHTTP] and additional requirements specific to the PSD2-Interface.

Elements of the "Signature" Header				
Element	**Type**	**Condition**	**Requirement [signHTTP]**	**Additional Requirement**
keyId	String	Mandatory	The keyId field is a string that the server can use to look up the component they need to validate the signature.	Serial Number of the TPP's certificate included in the "TPP-Signature-Certificate" header of this request. It shall be formatted as follows: keyId="SN=XXX,CA=YYYYYYYYYYYYYYYY" where “XXX" is the serial number of the certificate in hexadecimal coding given in the TPP-Signature-Certificate-Header and "YYYYYYYYYYYYYYYY" is the full Distinguished Name of the Certification Authority

Elements of the "Signature" Header				
Element	**Type**	**Condition**	**Requirement [signHTTP]**	**Additional Requirement**
				having produced this certificate.
Algorithm	String	Mandatory (Optional in **[signHTTP]**)	The "Algorithm " parameter is used to specify the digital signature algorithm to use when generating the signature. Valid values for this parameter can be found in the Signature Algorithms registry located at http://www.iana.org/assignments/signature-algorithms and MUST NOT be marked "deprecated". It is preferred that the algorithm used by an implementation be derived from the key metadata identified by the 'keyId' rather than from this field. […]The 'algorithm' parameter [..] **will most likely be deprecated in the future**.	Mandatory The algorithm must identify the same algorithm for the signature as described for the TPP's public key (Subject Public Key Info) in the certificate (Element "TPP-Signature-Certificate") of this Request. It must identify SHA-256 or SHA-512 as Hash algorithm.

Elements of the "Signature" Header				
Element	**Type**	**Condition**	**Requirement [signHTTP]**	**Additional Requirement**
Headers	String	Mandatory (Optional in **[signHTTP]**)	The "Headers" parameter is used to specify the list of HTTP headers included when generating the signature for the message. If specified, it should be a lowercased, quoted list of HTTP header fields, separated by a single space character. If not specified, implementations MUST operate as if the field were specified with a single value, the `Date` header, in the list of HTTP headers. Note that the list order is important, and MUST be specified in the order the HTTP header field-value pairs are concatenated together during signing.	Mandatory. Must include • "digest", • "x-request-id", Must conditionally include • "psu-id", if and only if "PSU-ID" is included as a header of the HTTP-Request. • "psu-corporate-id", if and only if "PSU-Corporate-ID" is included as a header of the HTTP-Request. • "tpp-redirect-uri", if and only if "TPP-Redirect-URI" is included as a header of the HTTP-Request. **No other entries may be included.** **Remark:** It is intended to introduce a new http header in a coming version. This new header shall indicate the creation date of a request on the side of the TPP. This new header and will also have to be included in this "Headers" element.

Elements of the "Signature" Header				
Element	**Type**	**Condition**	**Requirement [signHTTP]**	**Additional Requirement**
Signature	String	Mandatory	The "signature" parameter is a base 64 encoded digital signature, as described in RFC 4648 [RFC4648], Section 4. The client uses the `algorithm` and `headers` signature parameters to form a canonicalised `signing string`. This `signing string` is then signed with the key associated with `keyId` and the algorithm corresponding to `algorithm`. The `signature` parameter is then set to the base 64 encoding of the signature.	[No additional Requirements]

Example

Assume a TPP needs to include a signature in the following 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-ID:                 PSU-1234
PSU-User-Agent:         Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0
tpp-redirect-uri: https%3A%2F%2FshortURI_Cchallenge_Mmethod="S256"
Date:                   Sun, 06 Aug 2017 15:02:37 GMT

{
   "instructedAmount": {"currency": "EUR", "amount": "123"},
   "debtorAccount": {"iban": "DE23100100101234567 89"},
   "creditor": {"name": "Merchant123"},
   "creditorAccount": {"iban": "DE23100120020123456789"},
   "remittanceInformationUnstructured": "Ref Number Merchant"
}
```

So the body would encode to the following String in Base64:

eyAgICANCiAgICJpbnN0cnVjdGVkQW1vdW50IjogeyJjdXJyZW5jeSI6ICJFVVIiLCAiYW1vdW50IjogIjEyMyJ9LA0KICAgImRlYnRvckFjY291bnQiOiB7ImliYW4iOiAiREUyMzEwMDEwMDEwMTIzNDU2Nzg5In0sDQogICAiY3JlZGl0b3IiOiB7Im5hbWUiOiAiTWVyY2hhbnQxMjMifSwNCiAgICJcmVkaXRvckFjY291bnQiOiB7ImliYW4iOiAiREUyMzEwMDEyMDAyMDEyMzQ1Njc4OSJ9LA0KICAgInJlbWl0dGFuY2VJbmZvcm1hdGlvblVuc3RydWN0dXJlZCI6ICJSZWYgTnVtYmVyIE1lcmNoYW50Ig0KfQ==

and SHA-256 of the request body is

`KDUgmV/H0usna3yHPoXYteCFd1l32SWhOI45NTD0Ri4=` in Base64 (`'283520995FC7D2EB276B7C873E85D8B5E085775977D925A1388E393530F4462E'` in hexadecimal representation).

So using signature algorithm rsa-sha256 the signed request of the TPP will be

```
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-ID:                 PSU-1234
PSU-User-Agent:          Mozilla/5.0 (Windows NT 10.0; WOW64; rv:54.0)
Gecko/20100101 Firefox/54.0
tpp-redirect-uri: https%3A%2F%2FshortURI_Cchallenge_Mmethod="S256"
Date:                   Sun, 06 Aug 2017 15:02:37 GMT
Digest:                 SHA-
256=KDUgmV/H0usna3yHPoXYteCFd1l32SWhOI45NTD0Ri4=
Signature:              keyId="SN=9FA1,CA=CN=D-TRUST%20CA%202-1%202015,O=D-
Trust%20GmbH,C=DE",algorithm="rsa-sha256",
   headers="digest x-request-id psu-id tpp-redirect-uri",
   signature="Base64(RSA-SHA256(signing string))"
TPP-Signature-Certificate: TPP's_eIDAS_Certificate

{
   "instructedAmount": {"currency": "EUR", "amount": "123"},
   "debtorAccount": {"iban": "DE23100100101234567890"},
   "creditor": {"name": "Merchant123"},
   "creditorAccount": {"iban": "DE23100120020123456789"},
   "remittanceInformationUnstructured": "Ref Number Merchant"
}
```

Where *signing string* is

```
digest: SHA-256=KDUgmV/H0usna3yHPoXYteCFd1l32SWhOI45NTD0Ri4=
x-request-id: 99391c7e-ad88-49ec-a2ad-99ddcb1f7721
psu-id: PSU-1234
tpp-redirect-uri: https%3A%2F%2FshortURI_Cchallenge_Mmethod="S256"
```

NOTE: The header fields to be signed are denoted in small letters to clarify that the digest will use small letters for normalisation.

13 Requirements on the OAuth2 Protocol

The OAuth2 protocol as used optionally for this API is defined in [RFC6749]. In this section, additional requirements on the protocol are defined.

The requirements on the data exchange between the TPP and the OAuth Server of the ASPSP regarding the transport layer are identical to the data exchange requirements between TPP and the XS2A Interface, cp. Section 3.

Remark: Specifically, the requirements on using MTLS also apply to the usage of the oAUTH2 Protocoll. However, the general requirements on the application layer such as e.g. signing of Requests (see chapter 12) do not apply to the oAUTH2 messages.

The response type "code" and the grant types "authorization_code" and "refresh_token" are recommended by this specification. It is further strongly recommended to TPPs and ASPSPs to follow the security best practices defined in [OA-SecTop].

The ASPSP is required to provide TPPs with configuration data conforming to the "OAuth 2.0 Authorisation Server Metadata" specification.

13.1 Authorisation Request

For the "authorisation request" to the authorisation endpoint the following parameters are defined:

Query Parameters

Attribute	Condition	Description
response_type	Mandatory	"code" is recommended as response type.
client_id	Mandatory	organizationIdentifier as provided in the eIDAS certificate. The organizationIdentifier attribute shall contain information using the following structure in the presented order: - "PSD" as 3 character legal person identity type reference; - 2 character ISO 3166 country code representing the NCA country; - hyphen-minus "-" and - 2-8 character NCA identifier (A-Z uppercase only, no separator) - hyphen-minus "-" and - PSP identifier (authorization number as specified by NCA).

Attribute	Condition	Description
scope	Mandatory	PIS: The scope is the reference to the payment resource in the form "PIS:<paymentId>". AIS: The scope is the reference to the consent resource for account access in the form "AIS:<consentId>" PIIS: The scope is the reference to the consent resource for granting consent to confirmation of funds in the form "PIIS:<consentId>". Cancel-PIS: The scope is the reference to the payment cancellation in the form "Cancel-PIS:<paymentId>". **Note:** Cancel-PIS is an optional scope which the ASPSP can decide to use (whether it does must be specified in the ASPSP documentation). If they do so, it must also be used by the TPP for payment cancellation, and the PIS scope then refers only to payment authorisation.**Note:** The resource ids chosen by the ASPSP need to be unique to avoid resource conflicts during the SCA process.
state	Mandatory	A dynamical value set by the TPP and used to prevent XSRF attacks.
redirect_uri	Mandatory	the URI of the TPP where the OAuth2 server is redirecting the PSU's user agent after the authorization.
code_challenge	Mandatory	PKCE challenge according to cryptographic RFC 7636 (https://tools.ietf.org/html/rfc7636) used to prevent code injection attacks.
code_challenge_method	Optional	Code verifier transformation method, is "S256" or "plain". "S256" is recommended by this specification.

Example

```
GET /authorise?response_type=code&client_id="PSDES-BDE-3DFD21" &
```

```
scope=ais%3A1234-wertiq-983+offline_access&
state= S8NJ7uqk5fY4EjNvP_G_FtyJu6pUsvH9jsYni9dMAJw&
redirect_uri=https%3A%2F%2Fclient%2Eexample%2Ecom%2Fcb&
code_challenge_method="S256"
code_challenge=5c305578f8f19b2dcdb6c3c955c0aa709782590b4642eb890b97e43917cd
0f36 HTTP/1.1
    Host: api.testbank.com
```

13.2 Authorisation Response

The Authorisation Response of the ASPSP will deliver the following data:

Remark: As the request is not sent by the TPP but the PSU user agent, it will not be secured by the TPP's QWAC.

http Response Code

302

Query Parameters

Attribute	Condition	Description
Location:	Mandatory	redirect URI of the TPP
code	Mandatory	Authorisation code
state	Mandatory	Same value as for the request.

Example

```
HTTP/1.1 302 Found

Location: https://client.example.com/cb
?code=SplxlOBeZQQYbYS6WxSbIA
&state=S8NJ7uqk5fY4EjNvP_G_FtyJu6pUsvH9jsYni9dMAJw
```

13.3 Token Request

The TPP sends a POST request to the token endpoint in order to exchange the authorisation code provided in the authorisation response for an access token and, optionally, a refresh token. The following parameters are used:

Request Parameters

Attribute	Condition	Description
grant_type	Mandatory	"authorization_code" is recommended as response type.
client_id	Mandatory	cp. Definition in Section 13.1
code	Mandatory	Authorisation code from the authorisation response
redirect_uri	Mandatory	the exact uri of the TPP where the OAuth2 server redirected the user agent to for this particular transaction
code_verifier	Mandatory	PKCE verifier according to cryptographic RFC 7636 (https://tools.ietf.org/html/rfc7636) used to prevent code injection attacks.

Example

```
POST /token HTTP/1.1
Host: https://api.testbank.com
Content-Type: application/x-www-form-urlencoded
client_id="PSDES-BDE-3DFD21"
&grant_type=authorisation_code
&code=SplxlOBeZQQYbYS6WxSbIA
&redirect_uri= https%3A%2F%2Fclient%2Eexample%2Ecom%2Fcb
&code_verifier=7814hj4hjai87qqhjz9hahdeu9qu771367647864676787878
```

The TPP is authenticated during this request by utilising "Oauth 2.0 Mutual TLS Client Authentication and Certificate Bound Access Tokens" in conjunction with the TPP's eIDAS certificate.

13.4 Token Response

The ASPSPS responds with the following parameters:

Response Parameters

Attribute	Condition	Description
access_token	Mandatory	Access Token bound to the scope as requested in the authorisation request and confirmed by the PSU.
token_type	Mandatory	Set to "Bearer"
expires_in	Optional	The lifetime of the access token in seconds
refresh_token	Optional	Refresh Token, which can be utilised to obtain a fresh access tokens in case the previous access token expired or was revoked. Especially useful in the context of AIS.
scope	Mandatory	the scope of the access token

Example

HTTP/1.1 200 OK

```
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{
 "access_token": "SlAV32hkKG",
 "token_type": "Bearer",
 "expires_in": 3600,
 "refresh_token": "tGzv3JokF0XG5Qx2TlKWIA",
 "scope": "exampleScope"
}
```

13.5 Refresh Token Grant Type

The ASPSP may issue refresh tokens at its discretion, e.g. if an AISP uses the standard scope value "offline_access" or if the recurringIndicator in is set to true.

13.6 API Requests

When using the OAuth SCA approach, subsequent API requests are being authorized using the respective OAuth Access Token. The access token is sent to the API using the “Authorization” Header and the “BEARER” authorization schema as defined in RFC 6750.

This is an example API request

```
  GET /psd2/v1/payments/ sepa-credit-transfers/1234-wertiq-983/status
HTTP/1.1
  Host: https://api.testbank.com
  Authorization: Bearer SlAV32hkKG
```


14 Complex Data Types and Code Lists

In the following constructed data types are defined as used within parameter sections throughout this document.

14.1 PSU Data

Attribute	Type	Condition	Description
password	String	Conditional	Contains a password in plaintext.
encrypted Password	String	Conditional	Is used when a password is encrypted on application level.
additional Password	String	Conditional	Contains an additional password in plaintext
additional Encrypted Password	String	Conditional	Is provided when the additional password is used and is encrypted on application level.

14.2 TPP Message Information

Attribute	Type	Condition	Description
category	String	Mandatory	Only "ERROR" or "WARNING" permitted
code	Message Code	Mandatory	
path	String	Conditional	
text	Max500Text	Optional	Additional explaining text.

14.3 Amount

Attribute	Type	Condition	Description
currency	Currency Code	Mandatory	ISO 4217 Alpha 3 currency code

Attribute	Type	Condition	Description
amount	String	Mandatory	The amount given with fractional digits, where fractions must be compliant to the currency definition. Up to 14 significant figures. Negative amounts are signed by minus. The decimal separator is a dot. **Example:** Valid representations for EUR with up to two decimals are: • 1056 • 5768.2 • -1.50 • 5877.78

14.4 Address

Attribute	Type	Condition	Description
department	Max70Text	Optional	Identification of a division of a large organisation or building.
subDepartment	Max70Text	Optional	Identification of a sub-division of a large organisation or building.
streetName	Max70Text	Optional	Name of a street or thoroughfare.
buildingNumber	Max16Text	Optional	Number that identifies the position of a building on a street.
buildingName	Max35Text	Optional	Name of the building or house.
floor	Max70Text	Optional	Floor or storey within a building.
postBox	Max16Text	Optional	Numbered box in a post office, assigned to a person or organisation, where letters are kept until called for.
room	Max70Text	Optional	Building room number.
postCode	Max16Text	Optional	Identifier consisting of a group of letters and/or numbers that is added to a postal address to assist the sorting of mail.
townName	Max35Text	Optional	Name of a built-up area, with defined boundaries, and a local government. Usage Rule: If address lines are not used, this attribute is mandatory.
townLocationName	Max35Text	Optional	Specific location name within the town.
districtName	Max35Text	Optional	Identifies a subdivision within a country sub-division.
countrySubDivision	Max35Text	Optional	Identifies a subdivision of a country such as state, region, county.
country	Country Code	Mandatory	Nation with its own government.

14.5 Remittance

Attribute	Type	Condition	Description
reference	Max35Text	Mandatory	The actual reference.
referenceType	Max35Text	Optional	
referenceIssuer	Max35Text	Optional	

14.6 Links

The structure of Links is conform to [HAL].

Attribute	Type	Condition	Description
scaRedirect	href Type	Optional	A link to an ASPSP site where SCA is performed within the Redirect SCA approach.
scaOAuth	href Type	Optional	The link refers to a JSON document specifying the OAuth details of the ASPSP's authorisation server. JSON document follows the definition given in [RFC 8414].
confirmation	href Type	Optional	"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	href Type	Optional	A link to an endpoint, where the authorisation of a transaction or the authorisation of a transaction cancellation shall be started with a POST

Attribute	Type	Condition	Description
			command. No specific data is needed for this process start.
startAuthorisationWithPsuIdentification	href Type	Optional	The link to an endpoint where the authorisation of a transaction or of a transaction cancellation shall be started, where PSU identification shall be uploaded with the corresponding call.
updatePsuIdentification	href Type	Optional	The link to the payment initiation or account information resource, which needs to be updated by the PSU identification if not delivered yet.
startAuthorisationWithProprietaryData	hrefType	Optional	A link to the endpoint, where the authorisation of a transaction or of a transaction cancellation shall be started, and where proprietary data needs to be updated with this call. The TPP can find the scope of missing proprietary data in the ASPSP documentation. The usage of this hyperlink is not further specified in the specification but is used analogously to e.g. the startAuthorisation withPsuIdentification hyperlink.
updateProprietaryData	href Type	Optional	The link to the payment initiation or account information resource, which needs to be updated by the proprietary data.
startAuthorisationWithPsuAuthentication	href Type	Optional	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.
updatePsuAuthentication	href Type	Optional	The link to the payment initiation or account information resource, which needs to be updated by a PSU password and eventually the PSU identification if not delivered yet.
updateAdditionalPsuAuthentication	href Type	Optional	The link to the payment initiation or account information resource, which needs to be updated by an additional PSU password.

Attribute	Type	Condition	Description
startAuthorisationWithEncryptedPsuAuthentication	href Type	Optional	The link to an endpoint where the authorisation of a transaction or of a transaction cancellation shall be started, where encrypted PSU authentication data shall be uploaded with the corresponding call.
updateEncryptedPsuAuthentication	href Type	Optional	The link to the payment initiation or account information resource, which needs to be updated by an encrypted PSU password and eventually the PSU identification if not delivered yet.
updateAdditionalEncryptedPsuAuthentication	href Type	Optional	The link to the payment initiation or account information resource, which needs to be updated by an additional encrypted PSU password.
startAuthorisationWithAuthenticationMethodSelection	href Type	Optional	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	href Type	Optional	This is a link to a resource, where the TPP can select the applicable second factor authentication methods for the PSU, if there were several available authentication methods.
startAuthorisationWithTransactionAuthorisation	href Type	Optional	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.
authoriseTransaction	href Type	Optional	The link to the payment initiation or consent resource, where the "Transaction Authorisation"Request" is sent to. This is the link to the resource which will authorise the payment or the consent by checking the SCA authentication data within the Embedded SCA approach.

Attribute	Type	Condition	Description
self	href Type	Optional	The link to the payment initiation resource created by the request itself. This link can be used later to retrieve the transaction status of the payment initiation.
status	href Type	Optional	A link to retrieve the status of the transaction resource.
scaStatus	href Type	Optional	A link to retrieve the status of the authorisation or cancellation-authorisation sub-resource.
account	href Type	Optional	A link to the resource providing the details of one account
balances	href Type	Optional	A link to the resource providing the balance of a dedicated account.
transactions	href Type	Optional	A link to the resource providing the transaction history of a dedicated account.
cardAccount	href Type	Optional	A link to the resource providing the details of one card account.
cardTransactions	href Type	Optional	A link to the resource providing the transaction history of a dedicated card account.
transactionDetails	href Type	Optional	A link to the resource providing details of a dedicated transaction.
first	href Type	Optional	Navigation link for paginated account reports.
next	href Type	Optional	Navigation link for paginated account reports.
previous	href Type	Optional	Navigation link for paginated account reports.
last	href Type	Optional	Navigation link for paginated account reports.
download	href Type	Optional	Download link for huge AIS data packages.

14.7 href Type

Attribute	Type	Condition	Description
href	String	Mandatory	

14.8 Authentication Object

Attribute	Type	Condition	Description
authenticationType	Authentication Type	Mandatory	Type of the authentication method.
authenticationVersion	String	Conditional	Depending on the "authenticationType". This version can be used by differentiating authentication tools used within performing OTP generation in the same authentication type. This version can be referred to in the ASPSP's documentation.
authenticationMethodId	Max35Text	Mandatory	An identification provided by the ASPSP for the later identification of the authentication method selection.
name	String	Mandatory	This is the name of the authentication method defined by the PSU in the Online Banking frontend of the ASPSP. Alternatively this could be a description provided by the ASPSP like "SMS OTP on phone +49160 xxxxx 28". This name shall be used by the TPP when presenting a list of authentication

Attribute	Type	Condition	Description
			methods to the PSU, if available.
explanation	String	Optional	detailed information about the SCA method for the PSU

14.9 Authentication Type

More authentication types might be added during implementation projects and documented in the ASPSP documentation.

Name	Description
SMS_OTP	An SCA method, where an OTP linked to the transaction to be authorised is sent to the PSU through a SMS channel.
CHIP_OTP	An SCA method, where an OTP is generated by a chip card, e.g. an TOP derived from an EMV cryptogram. To contact the card, the PSU normally needs a (handheld) device. With this device, the PSU either reads the challenging data through a visual interface like flickering or the PSU types in the challenge through the device key pad. The device then derives an OTP from the challenge data and displays the OTP to the PSU.
PHOTO_OTP	An SCA method, where the challenge is a QR code or similar encoded visual data which can be read in by a consumer device or specific mobile app. The device resp. the specific app than derives an OTP from the visual challenge data and displays the OTP to the PSU.
PUSH_OTP	An OTP is pushed to a dedicated authentication APP and displayed to the PSU.
SMTP_OTP	An OTP is sent via email to the PSU.