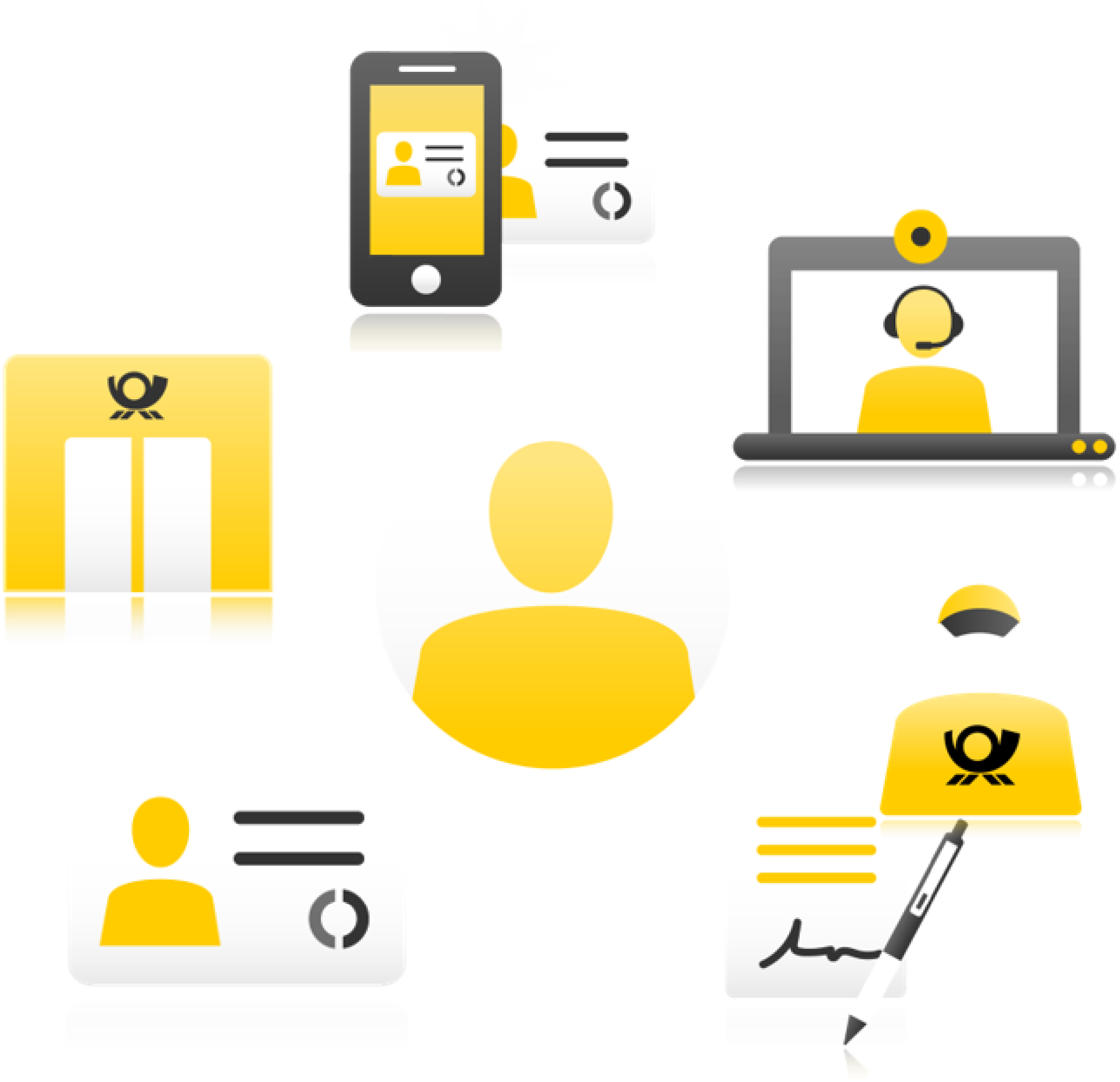
POSTIDENT

SCR-Signing API Guide



Version 1.3
Stand 15.08.2024

Table of Contents

Changelog

15.08.2024	Identification with Online ID function added
12.08.2021	HTTPs is now required for webhook URLs
12.01.2021	• Clarification on the type of the PDF form field specified in signatureFieldName • Clarification on the cardinalitites of result data • Salutation no longer filled (Rel 6.4), field salutation will be removed (Rel. 6.5)
07.10.2020	New webhook source IP address 156.137.9.65
08.06.2020	Clarification on value "unchecked" in JSON Entity "ResultValue"
25.03.2020	Description of JSON structure for status data modified
07.03.2019	Information for email communication by the client added
02.03.2020	Differentiated between HTTP error 401 and 403
05.02.2019	SigningCaseResult: new values 6 - 12 for Identification Document "type" in the description added Process description with client and partner added
13.12.2018	Retrieve status of a signing case added
10.09.2018	SignerNumber to POST example added
06.08.2018	• "Signer.SignedDocuments" indexes to being 1-based corrected • descriptions for substatus added
19.07.2018	Webhook source IPs added
16.01.2018	Description how to leave not mandatory fields added
01.01.2018	V2 created

1 Overview

The POSTIDENT Standard Connect & Results for Signing API (SCR-Signing) enables you to start a new signing case and fetch the results. With the SCR-Signing V2 API you can create a signing case with one or multiple signers (the SCR-Signing V1 is restricted to one signer in one signing case).

1.1 Definitions

English	German	Description
client	Geschäftskunde, Auftraggeber	Business customer which is the principal for the identification process
clientId	Client ID	Identifier for a client to get access to the SCR-Signing API. Each client ID has a configuration that controls the behaviour of the associated signing cases.
ident case	Identifizierungsvorgang	Container for a signing identification process; may contain one or more identifications of a user
signing case	Signaturvorgang	Container for a signing identification process; may contain one or more identifications of a user
identification	Identifikation	Identification using a POSTIDENT identification method
signer, user	ZiP, Endkunde, PK, Unterzeichner	Private customer who wants to sign the documents

1.2 Preconditions

During setup you should have received data:

- clientId
- username and password for the REST API (required for authentication)
- data password (required for decryption of result data)

1.3 API calls to implement

- **SCR-Signing POST to start a new signing case**
  With the SCR-Signing POST you transfer the PDF documents to be signed with the personal data of the user(s) who wants to sign the documents.
  for more details see 4. REST API-Start a new signing case

- **SCR-Signing GET to retrieve signing result**
  With the SCR-Signing GET you get the signed documents and data of the identified user(s). Pictures of the identification documents and the identification result PDF are **not** included in the response of SCR-Signing GET.
  for more details see 6. REST API - Retrieve a signing case result

- **SCR-Ident GET to retrieve detailed ident results ⚠ mandatory for GWG clients (clients whose business is subject to the Money Laundering Act)**
  With the SCR-Ident GET you get extended information related to the identification part of the process, including the pictures of the identification documents and the identification result PDF.
  for more details see 7. REST API - Retrieve detailed ident results

2 Technical Details

2.1 Protocol

HTTPS is used to ensure that all parameters are encrypted.

2.2 Host

Environment	URL	Remarks
integration	on demand	
productive	postident.deutschepost.de	

2.3 Path and Versioning

All REST endpoints contains the following elements:

Element	Description	Example
version	Use „v2“	v2

clientid	Provided by Deutsche Post. Uniquely identifying your access to the API. Different clientIds can e.g. be used to choose between different sets of configuration settings. **Format**: alphanumeric, uppercase (case sensitive).	1234ABCD

Example:

```
POST /api/scr-signing/v2/1234ABCD/signingcases
```

2.4 Authentication

HTTPS and Basic Auth (RFC 2617) are used for the authentication. Username and password must be transmitted in the HTTP header according to Basic Auth.

2.5 Header

Element	**Mandatory**	**Description**	**Example**
Content-Type	Yes	Use "application/json"	application/json
Authorization	Yes	Basic <encoded username and pw>	Basic U0NSV1VMOlFqVk53d1QzUjgodQ==

2.6 Body

2.6.1 UTF-8 encoded JSON

The content has to be sent as UTF-8 encoded JSON.

Following characters are not allowed:

<

>

&lt;

&gt;

&#60;

&#62;

&#x3c;

&#x3e;

{

}

&#123;

&#125;

&#x7b;

&#x7d;

2.6.2 JSON structure

The JSON structures are defined in 10. REST API - JSON structures.

This specification documents all parameters, data structures, required fields, field types and the maximum field lengths.

2.6.3 Leaving not mandatory fields

Fields that are not mandatory can be left by not adding the keys to the JSON.

Example given:

```
{"field1":"value",
 "notMandatoryField": "value
}
```

If the field "notMandatoryField" is not mandatory it can be left as given in the following JSON:

```
{"field1":"value"}
```

The signing systems distinguishes between missing values, that are left out of the JSON, and empty values. If an empty field is filled with a value during the identification step, this field is marked as "changed" in the result query. If this happens for a field with a missing value, the field is marked as "new".

3 REST API - General Flow

You can create a signing case with one or multiple users to sign the documents.

The email communication with the signers can either by carried out by you or by the Signing System. This behaviour can be configured in the FA-Portal.

3.1 General Flow for one Signer

SigningCase Creation:

- Your web portal collects personal and contractual information from the signer in your business process.
- Your web portal generates the PDF documents to be signed with the personal data of the signer.
- Your server creates a unique referenceId (optional).
- Your server POSTs known personal data of the signer and the to be signed PDF documents to the POSTIDENT E-Signing server, which validates the received data including the PDF documents. In return your server gets a caseId plus a caseURL and a resume URL. The resume URL is needed for email communication with the signer. The state of the signing case is "signing process", while all signers have the state "new".

Redirect to POSTIDENT E-Signing web portal

- Then you have two options:
    a. Immediate start: Your web portal redirects the signer to his start URL. See Redirect the User to the POSTIDENT E-Signing Portal for more information.
    b. Delayed start respectively resumption of the case after a break: An email is send to the signer with the resume URL. The signing system will send this mail if configured. Otherwise your server has to sent a mail to the signer. This mail must contain the resume URL. The signer starts the signing process in the POSTIDENT E-Signing web portal with this resume URL.
- The signer gets an overview of the complete signing process un the POSTIDENT E-Signing web portal.

Identification via Videochat or with Online ID function:

- The signer is identified by an agent via video identification or with Online ID function. After successful identification the signing case has the state "signing process". If you are interested in the identification results independent of the signature success (for example if the results can be used for your KYC process), then you should persist the identification data now.
- You can request the identification result via **SCR-Ident API** with the same credentials using your clientId and identCaseId.

Signing Documents:

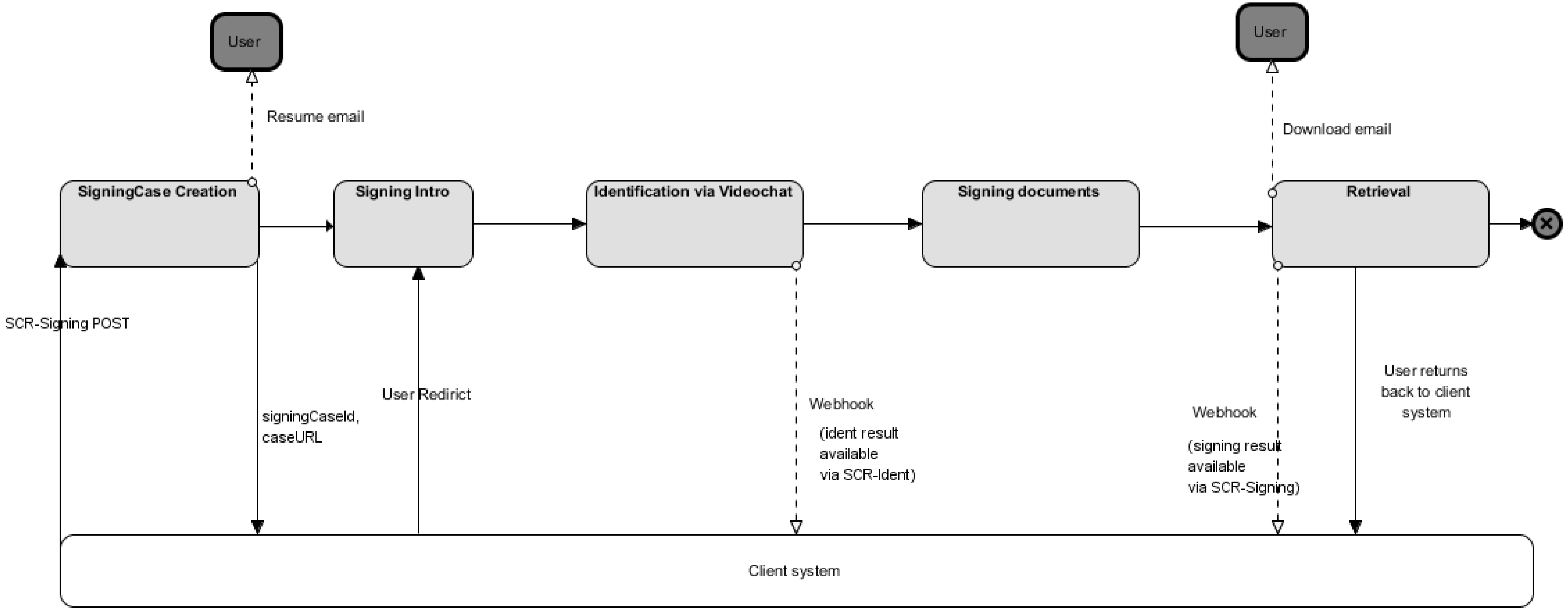
- The signer previews the documents.

- The signer signs the documents using SMS-TAN. The signing case receives the state "signed".

Retrieval

- Signer downloads the signed PDF documents and will be redirected to your web portal.
- Your system queries the identity data and the signed PDF documents from the POSTIDENT E-Signing server via **SCR-Signing API.**

- During the retrieval period, starting with the signing of the documents, your system can query identity data via **SCR-Ident API** (including pictures of the identification documents and the
identification result PDF) and the signed PDF documents via **SCR-Signing API**. POSTIDENT E-Signing server sends a link via email to the signer to download the signed PDF documents in the POSTIDENT E-Signing web portal. The duration of the retrieval period is configured as retention time in your business configuration.
- After the retreval period the signing case gets the state "closed" and the signing case including the PDF documents and identity data will be deleted. Audit logs about the signing case will be archived.



3.2 General Flow for two Signers

You can create a signing case with multiple signers to sign PDF documents. In this case the signers will run through the steps identification and signing documents in a sequence (in the same order you provide the signers in the POST).

The following part describes only the differences between a signing case with two signers or one signer (described above).

SigningCase Creation:

- Your POST includes an array of signers containing signers' data and in response you get a caseURL and a resume URL for each signer. You can specify which PDF document each signer has to sign.

Redirect to POSTIDENT E-Signing web portal

- You redirect the first signer to the POSTIDENT E-Signing web portal or send an email with the resume URL to the first signer to start the signing process.
- POSTIDENT E-Signing server sends an email to the first signer with the resume URL.

First Signer: Identification via Videochat or with Online ID function, Signing Documents

- After signing the PDF documents the first signer will be redirected to your web portal. PDF Documents signed just by the first signer can **not** be retrieved by the first signer.

Second Signer: Identification via Videochat or with Online ID function, Signing Documents

- When the first signer has signed the PDF documents POSTIDENT E-Signing server sends an email to the second signer to start the signing process.
- The second signer signs and downloads the PDF documents and will be redirected to your web portal.

Retrieval

- POSTIDENT E-Signing server sends an email to both signers with a link for the download of the signed PDF documents.

3.3 Exception Flows

- In the case that the signer is declined, the POSTIDENT E-Signing server will notify the signer via email.
- Signing case with multiple signers: if one of the signers has been declined the whole signing case will be declined. All signers will be notified via email.
- In the case that the postprocess of the identification takes longer, the signer will be informed about the delay. After successful identification the signer will be notified via email to continue the signing process. In the case of a declined identification, the signer will be also notified via email. If the email communication by the E-Signing system is deactivated for your account, you have to notify the signers. In this case the value of the parameter signer.notifyUserAboutSigningState will be "true".

3.4 Status Overview

3.4.1 Signing Case Status Overview

A signing case can have these status values during its lifecycle.

Status	**Description**	**E-Mail by E-Signing system**[1)]	**Web-hook**[2)]
new	Signing case has been created		
signing processs	Signing process includes identification and signing documents by the signer(s) of the signing case		
signed	All documents have been signed by all signers of the signing case	x	x
closed	The signing case has been closed after the retrieval period. The documents can no longer be accessed by client or signers		

<table>
<tr><th>Status</th><th>Description</th><th>E-Mail by E-Signing system[1]</th><th>Web-hook[2]</th></tr>
<tr><td>declined</td><td>The signing case has been declined, for example due to fraud suspicion. If one signer has been declined the whole signing case will be declined
Substatus describes the declined status on a more detailed level:
<table>
<tr><th>Substatus</th><th>Description</th></tr>
<tr><td>validity period expired</td><td>The validity of the case, specified on creation in the field "validUntil", has expired without a successful signature.</td></tr>
<tr><td>signer was declined</td><td>One of the signers was declined. The case could therefore not be finished successfully. For the root cause, see the field "signerSubstatus" in the Signer result data.</td></tr>
</table></td><td>x</td><td>x</td></tr>
</table>

1) Can be deactivated for your account (= clientId).

2) Must be activated for your account (= clientId)

3.4.2 Signer Status Overview

A signing case has one or more signers. Every signer has his own status model.

<table>
<tr><th>Status</th><th>Description</th><th>E-Mail by E-Signing system[1]</th><th>Web-hook[2]</th><th>Redirect (Callback-URL)[3]</th></tr>
<tr><td>new</td><td>Signer has been created</td><td>x</td><td></td><td></td></tr>
<tr><td>identification</td><td>The signer is currently in the identification process
<table>
<tr><th>Substatus</th><th>Description</th></tr>
<tr><td>prolongued duration of identification postprocess</td><td>The postprocessing of the identification takes longer then expected, for example due</td></tr>
</table></td><td></td><td></td><td></td></tr>
<tr><td>signing process</td><td>The person has been identified and is now in the process of viewing and signing the documents
(if the parameter signer.notifyUserAboutSigningState is true, signer gets an email after successful or declined identification)</td><td>(x)</td><td>x</td><td></td></tr>
<tr><td>signed</td><td>All documents have been signed by this signer</td><td>x</td><td></td><td>x</td></tr>
<tr><td>finished</td><td>The retreval period is over. The signer has finished the signing process. The documents can no longer be accessed by signer or client</td><td></td><td></td><td></td></tr>
</table>

<table>
<tr><th>Status</th><th>Description</th><th>E-Mail by E-Signing system[1]</th><th>Web-hook[2]</th><th>Redirect (Callback-URL)[3]</th></tr>
<tr><td>declined</td><td>The signing case has been declined, for example due to fraud suspicion.
Substatus describes the declined status on a more detailed level:
<table>
<tr><th>Substatus</th><th>Description</th></tr>
<tr><td>signing declined</td><td>The signing case was declined. For the root cause, see the field "caseSubstatus" in the signing case result data.</td></tr>
<tr><td>ident declined</td><td>The identification was declined, for example due to fraud suspicion. For more details query the identification result via SCR-Ident API.</td></tr>
<tr><td>mobile phone number not verified</td><td>The mobile phone number of the signer could not be verified during the identification process via a SMS-TAN, but instead the TAN was send by E-Mail. This should happen only when it wasn't possible to send the SMS on multiple tries and after the signer has been informed that a signature is not possible after switching the TAN channel to E-Mail.
Since the mobile phone number is required as strong authentication factor for the access of the signer to its certificate, it wasn't possible to proceed with the signature.</td></tr>
</table></td><td>x</td><td></td><td></td></tr>
</table>

<table>
<tr><th>Status</th><th>Description</th><th>E-Mail by E-Signing system[1)]</th><th>Web-hook[2)]</th><th>Redirect (Callback-URL)[3)]</th></tr>
<tr><td></td><td>
<table>
<tr><th>Substatus</th><th>Description</th></tr>
<tr><td>mobile phone number not unique</td><td>Only relevant for cases with multiple signers: For security reasons and to establish "non-repudiation" each signer must have a unique mobile phone number. If the phone number was changed during the identification so that multiple signers have the same number, the case gets declined.</td></tr>
<tr><td>maximum number of started operations exceeded</td><td>The number of times a signer can launch the signing process is limited for security reasons. This maximum was exceeded.</td></tr>
<tr><td>maximum number of created certificates exceeded</td><td>If a signer interrupts the signing process after the issuance of the certificate and continues later (after the short lived certificate is expired), then a new certificate must be issued. The number of certificates issued for the same signer is therefore limited. This maximum was exceeded.</td></tr>
</table>
</td><td></td><td></td><td></td></tr>
</table>

<table>
<tr><th>Status</th><th>Description</th><th>E-Mail by E-Signing system[1]</th><th>Web-hook[2]</th><th>Redirect (Callback-URL)[3]</th></tr>
<tr><td></td><td><table>
<tr><th>Substatus</th><th>Description</th></tr>
<tr><td>maximum
number of
sent sms exceeded</td><td>The maximum number of SMS-TAN sent to the signer for authentication purposes is limited for security reasons. This maximum was exceeded.</td></tr>
<tr><td>other signer
was declined</td><td>Only relevant for cases with multiple signers:
This signer had already done a successful signatur but one of the succeeding signers was declined</td></tr>
</table></td><td></td><td></td><td></td></tr>
</table>

1) Can be deactivated for your account (= clientId)

2) Must be activated for your account (= clientId)

3) For use in return button back to client

3.5 E-Mail Communication

By default, the POSTIDENT E-Signing system informs the user on several occasions via email, see the Chapters "General Flow for one Signer" and "Statuses of a Signing Case and a Signer" above.

In case you want to take over the email communication by yourself, your account can be configured to suppress these emails. In this case you have to send the emails to the signers.
The emails you should sent are described below. You get all necessary information via SCR-Signing GET method.

E-Mail name	E-Mail content/objective	Recipients	State/Trigger
resume	Information for the signer how to resume the signing process after a break. Resumption link (signers[0].webStart.resumeCaseURL) to resume the signing process	First signer	Directly after the creation of the case when case status is "signing process": Signing case status "signing process" Signer status "new"
start signing (subsequent signer)	Link to start the signing process as the next signer of a case with multiple signers. Resumption link (signers[x].webStart.resumeCaseURL) to resume the signing process	Next signer with status = "new"	Directly after the previous signer has signed the documents Signing case status "signing process" Preceding signer status changed from "signing process" to "signed"
ident reminder	Remind active signer of finishing the identification Resumption link (signers[x].webStart.resumeCaseURL) to resume the signing process	Signer with status = "identification"	The signer has been in status "identification" for 24 h and the case is at least valid for the next 2 hours The signer has been in status "identification" for 48 h and the case is at least valid for the next 2 hours Signing case status "signing process" Signer status "identification"

E-Mail name	E-Mail content/objective	Recipients	State/Trigger
signing reminder	Remind active signer of finishing the signing Resumption link (signers[x].webStart.resumeCaseURL) to resume the signing process	Signer with status = "signing process"	The signer has been in status "signing process" for 48 h and at least valid for the next 2 hours Signing case status "signing process"
download	Inform the signer where he can (re-)download the signed documents: Resumption link (signers[x].webStart.resumeCaseURL) to resume the download of the signed documents	All signers	Signing case status changed from "signing process" to "signed"
declined	Inform signers that the digital signature wasn't possible	All signers	Signing case status changed from "signing process" to "declined" and substatus != "validity period expired"
expired	Inform signers that the validity of the signing case expired	All signers	Signing case status changed from "signing process" to "declined" and substatus = "validity period expired"

3.6 Push-Notifications via Webhook

You can receive a push notification for specified status changes of a signing case, for more details see 9.Webhook.

The Webhook feature must be activated for your account.

3.7 Redirect/Callback URLs

With this feature you can redirect the user to a custom URL on your web server after the signing process is finished.

You can provide the CallbackURLs in your initial call:

- Parameter `callbackUrlDeclined`: Used after signing was declined.
- Parameters `callbackUrlSuccess`: Used after signing the documents by the user.

4 REST API - Start a new signing case

4.1 Path

Start a new signing process with a POST request using the following URL:

`/api/scr-signing/{version}/{clientId}/signingcases`

4.2 Functionality

The service

- validates the input data
- checks the compliance of the PDF documents with PDF/A-2b and converts them if possible (see SCR-Signing API Guide - PDF document requirements for more details).
    - If a document is not compliant to PDF/A-2 and the compliance can not be achieved by an automatic conversion, then the service request fails with an error and returns details about the cause of the problem
- persists a signing case and returns the unique signing case id: caseId

4.3 Sample Request

```
POST <host>/api/scr-signing/v1/1234ABCD/signingcases

Content-Type: application/json
Authorization: Basic <encoded username and pw>
Accept: application/json

{
  "processData": {
    "caseName": "Ratenkredit Musterbank",
    "targetCountry": "DEU",
    "preferredLanguage": "DE_DE",
    "webHookUrl": "https://musterbank.eu/api/pi-signing/webhooklistener",
    "referenceId": "V100012345",
    "validUntil": "2017-02-20",
    "phoneNumberClientCustomerService": "+49 228 123456789",
    "callbackUrlSigningDeclined": {
      "webUrl": "https://musterbank.eu/pi-signing-callback/declined.html"
    }
  },
  "documents": [
    {
      "name": "Ratenkredit",
      "referenceId": "V100012345-01",
      "hasToBeSigned": true,
      "mimeType": "application/pdf",
      "fileName": "V100012345-01.pdf",
      "documentData": "TWFuIG...VyZS4="
    }
  ],
  "signers": [
   {
      "contactData": {
        "mobilePhone": "+49171123456789",
        "email": "erika.mustermann@internet.de"
      },

      "identityData": {
        "firstName": "Erika",
        "lastName": "Mustermann",
        "birthName": "Meier",
        "birthDate": "1964-08-12",
        "birthPlace": "Berlin",
        "nationality": "DEU",
        "address": {
          "city": "Köln",
          "streetAddress": "Heidestr. 17",
          "postalCode": "51147",
          "country": "DEU"
        }
      },
    "documentsToSign": [
     {
        "documentToSign": 1,
          "signatureStampPosition": {
            "pageNumber": 1,
            "left": 100,
            "top": 500
          }
       }
       ],
      "callbackUrlSigningSuccess": {
        "webUrl": "https://musterbank.eu/pi-signing-callback/success.html"
      }
    },
   {
      "contactData": {
        "mobilePhone": "+49171123456710",
        "email": "max.mustermann@internet.de"
      },

      "identityData": {
        "firstName": "Max",
        "lastName": "Mustermann",
        "birthDate": "1965-10-15",
        "birthPlace": "Hamburg",
        "nationality": "DEU",
        "address": {
          "city": "Köln",
          "streetAddress": "Heidestr. 17",
          "postalCode": "51147",
          "country": "DEU"
        }
      },
    "documentsToSign": [
     {
        "documentToSign": 1,
          "signatureStampPosition": {
            "pageNumber": 1,
            "left": 200,
            "top": 500
          }
       }
       ],
      "callbackUrlSigningSuccess": {
        "webUrl": "https://musterbank.eu/pi-signing-callback/success.html"
      }
    }
   ]
  }
```

4.4 Response

HTTP status codes in the response for success:

Http Status Code	Message	Possible Cause
201	Created	The signing case was successfully created

You will also get application/JSON containing the following information:

Response example:

```
{
  "caseId": "YR9W91GEZK24",
  "signers: [
    {
      "signerNumber": 1,
      "webStart": {
        "caseURL": "https://postident.deutschepost.de/signingportal/entry/e29c1298-5bff-4b0e-aa53-30b93d533840",
        "resumeCaseURL": "https://postident.deutschepost.de/signingportal/reentry/j67s1594-5bff-4b0e-aa53-30b93d533840",
        "redirectTokenSecret": "icEfRPW4exlKe0nDsXSHoyk7uQpupdFaFwWyT1Z8Ub8="
      }
    },
    {
      "signerNumber": 2,
      "webStart": {
        "caseURL": "https://postident.deutschepost.de/signingportal/entry/e29c1298-5bff-4b0e-aa53-30b93d533940",
        "resumeCaseURL": "https://postident.deutschepost.de/signingportal/reentry/e29c1298-5bff-4b0e-aa53-30b93d533940",
        "redirectTokenSecret": "isAfeWO4exlKe0nDsXSHoyk7uQpupdFaFwWyT1Z8Ub8="
      }
    }
 ]
}
```

4.5 Errors

HTTP status codes in the response for error situations:

Http Status Code	**Message**	**Possible Cause**
400	Bad request	Invalid order data, e.g. • JSON agreement violated • Mandatory field missing • Field wrongly formatted • Field value too long See the „key“ field of the error json to see which field is wrong
401	Unauthorized	Wrong or missing credentials, e.g. - Wrong or missing username or password
403	Forbidden	Missing authorization, e.g. • user not authorized for the endpoint • user not authorized for the clientId

The Response body will contain additional information regarding the error. See below some examples.

4.5.1 Example 1: Badly formatted email field

```
HTTP status code: 400
```

```
{
    "apiVersion": "v1",
    "errors": [
      {
        "reason": "incorrect value",
        "errorcode": "90005",
        "key": "contactData.email",
        "message": "The provided value is not valid."
      }
    ]
}
```

4.5.2 Example 2: Badly formatted date field

```
HTTP status code: 400
{
    "apiVersion": "v1",
    "errors": [
      {
        "reason": "incorrect value",
        "errorcode": "90001",
        "key": "identityData.dateIssued",
        "message": "Date field format must be YYYY-MM-DD."
      }
    ]
}
```

4.5.3 Example 3: Missing or wrong authorization

```
HTTP status code: 401

No additional information due to security reasons.
```

5 Redirect the User to the POSTIDENT E-Signing Portal

For the redirection of the user to the POSTIDENT E-Signing portal you must use a HTTP form post to the URL received on signing case creation in the response field "webStart.caseURL". This post must comprise a so called "redirect token". The token carries a signature to prove the authenticity of your system and thus allows the redirected user to access the sensitive data of the signing case.

5.1 Creation of the Redirect Token

The redirect token is following the JSON Web Token (RFC 7519) standard. Libraries for the issuing of such tokens are available for most programming languages.

This token must be signed with the HMAC-SHA256" (HS256) algorithm and use the value from the response field "webStart.redirectTokenSecret" as secret key.

Use these values to fill the fields of the token

Field	Description
sub	<Case Id of the signing case>
iss	<Client ID>
aud	"Signing"
exp	This field is not validated in the current state.

5.1.1 Example Token:

The following example shows the content of a JSON Web Token before it is signed.

JWT JSON example (without signature)

```
{
  "aud": "Signing",
  "sub": "M3FB00URX4A3",
  "iss": "C969BCE4",
  "exp": 1504018064
}
```

Base64 encoded representation of the signed token:

"eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJTaWduaW5nIiwic3ViIjoiTTNGQjAwVVJYNEEzIiwiaXNzIjoiQzk2OUJDRTQiLCJleHAiOjE1MDQwMTgwNjR9.iWAnMo-hoo-u6VcQaupMvIgHQqQ4dLyG74C68iMxJbA"

5.1.2 Code sample for creating a JWT token:

Find below an (pseudo code) example using the java library "Nimbus DS":

Issue Redirect Token - Java Pseudo Code

```
import com.nimbusds.jose.JOSEException;
import com.nimbusds.jose.JWSAlgorithm;
import com.nimbusds.jose.JWSHeader;
import com.nimbusds.jose.JWSSigner;
import com.nimbusds.jose.crypto.MACSigner;
import com.nimbusds.jwt.JWTClaimsSet;
import com.nimbusds.jwt.SignedJWT;

...
```

```
    Calendar expirationTime = Calendar.getInstance();
    expirationTime.add(Calendar.MILLISECOND, 5 * 60 * 1000);

    // Prepare Token
    JWTClaimsSet claimsSet = new JWTClaimsSet.Builder()
            .audience("Signing")
            .issuer("FEBFD743")
            .subject(createCaseResponse.getCaseId())
            .expirationTime(expirationTime.getTime())
            .build();

    // Create HMAC signer
    byte[] secretAsByteArray = Base64.getDecoder().decode(createCaseResponse.getRedirectTokenSecret());
    JWSSigner signer = new MACSigner(secretAsByteArray);
    SignedJWT signedJWT = new SignedJWT(new JWSHeader(JWSAlgorithm.HS256), claimsSet);
    signedJWT.sign(signer);
    String tokenString = signedJWT.serialize();
```

5.2 Testing of the Token

For test purposes you can validate the token with the JWT Token Debugger. If you want to verify the signature, paste value from the field "webStart.redirectTokenSecret" as received from the SCR-Signing API in the input box "secret" and check the box "secret base64 encoded".

5.3 Posting of the Token

The actual redirection can be executed by a self posting HTML form. Place the token as HTTP Post parameter in the parameter "authToken".

```
<!DOCTYPE html>
<html>
```

```
    <body onload="javascript:document.forms[0].submit()">
        <form method="post" action="https://postident.deutschepost.de/signingportal/entry/e29c1298-5bff-4b0e-aa53-30b93d533840" >
            <input type="hidden" name="authToken" value="eyJhbGciOiJIUzI1NiIsInR5cCI6IkpX.....0RMHrHDcEfxjoYZgeFONFh7HgQ"/>
        </form>
    </body>
</html>
```

6 REST API - Retrieve a signing case result

6.1 Path

Request the current status and the results of a signing case.

```
GET /api/scr-signing/{version}/{clientId}/signingcases/{caseId}
```

Parameter	Default	Description
`includeBinaryData`	`false`	Toggles inclusion of BASE64 encoded data for binary objects, signed documents.

6.2 Retrieval period

Retrieval period starts with the signing of the documents. During the retrieval period your system can query identity data and the signed documents. The user can access the signed documents in the E-Signing portal. The duration of the retrieval period is configured as retention time in your business configuration. After the retreval period, the signing case and the signed documents will be deleted automatically.

6.3 Sample Request and Response

Request to retrieve a single signing case by case ID:

```
GET /api/scr-signing/v1/1234ABCD/signingcases/YR9W91GEZK24 HTTP/1.1
Host: postident.deutschepost.de
```

```
Authorization: Basic R0s0Mi5TQ1I6cEpzZW43NWh3biF0
```

Response for successful signing case:

```
{
  "caseId": "YR9W91GEZK24",
  "referenceId": "V100012345",
  "caseStatus": "signed",
  "validUntil": "2017-02-20",
  "storeUntil ": "2017-05-"29",
  "signedDocuments": [
    {
      "referenceId": "V100012345-01",
      "mimeType": "application/pdf",
      "fileName": "V100012345-01.pdf",
      "documentData": "ABFuIG...VkPcD3="
    }
  ],
  "signers": [
   {
      "signerNumber": 1,
      "identCaseId": "1BC7FG34LK09",
      "created": "2017-01-31T13:30:14.869Z",
      "modified": "2017-01-31T13:35:14.869Z",
      "identificationTime": "2017-01-31T13:40:14.869Z",
      "identificationMethod": "video",
      "signingTime": "2017-01-31T13:40:14.869Z"
      "termsAndConditionsAcceptedTime": "2017-01-31T13:39:14.869Z"
      "signerStatus": "signed",
      "signerContactData": {
         "mobilePhone": {
           "status": "match",
           "value": "+49171123456789"
         },
         "email": {
           "status": "unchecked",
           "value": "erika.mustermann@internet.de"
         }
      },
      "identityData": {
        "firstName": {
          "status": "match",
          "value": "Erika"
        },
        "lastName": {
          "status": "match",
          "value": "Mustermann"
        },
        "birthName": {
          "status": "new",
          "value": "Müller"
         },
        "birthDate": {
          "status": "new",
          "value": "1964-08-12"
        },
        "birthPlace": {
          "status": "new",
          "value": "Berlin"
        },
        "nationality": {
```

```
        "status": "new",
        "value": "DEU"
      },
      "address": {
        "streetAddress": {
          "status": "match",
          "value": "Heidestr. 17"
        },
        "postalCode": {
          "status": "match",
          "value": "51147"
        },
        "city": {
          "status": "match",
          "value": "Köln"
        },
        "country": {
          "status": "match",
          "value": "DEU"
        }
      }
    },
    "identificationDocument": {
      "type": "1",
      "number": "O8154711XT9",
      "dateIssued": "2010-05-20",
      "dateOfExpiry": "2020-05-19",
      "authority": "Landeshauptstadt Berlin",
      "placeOfIssue": "Berlin",
      "countryOfDocument": "DEU"
    },
    "signedDocuments": [
      1
    ]
  },
  {
    "signerNumber": 2,
    "identCaseId": "3AS5CV34LK09",
    "created": "2017-02-01T13:30:14.869Z",
    "modified": "2017-02-01T13:35:14.869Z",
    "identificationTime": "2017-02-01T13:40:14.869Z",
    "identificationMethod": "video",
    "signingTime": "2017-02-01T13:40:14.869Z"
    "termsAndConditionsAcceptedTime": "2017-02-01T13:39:14.869Z"
    "signerStatus": "signed",
    "signerContactData": {
       "mobilePhone": {
         "status": "match",
         "value": "+49171123456987"
       },
       "email": {
         "status": "unchecked",
         "value": "max.mustermann@internet.de"
       }
    },
    "identityData": {
      "firstName": {
        "status": "match",
        "value": "Max"
      },
      "lastName": {
        "status": "match",
        "value": "Mustermann"
      },
      "birthDate": {
        "status": "new",
        "value": "1966-02-15"
      },
      "birthPlace": {
        "status": "new",
        "value": "Berlin"
      },
      "nationality": {
        "status": "new",
        "value": "DEU"
      },
      "address": {
        "streetAddress": {
          "status": "match",
          "value": "Heidestr. 17"
        },
        "postalCode": {
          "status": "match",
          "value": "51147"
        },
        "city": {
          "status": "match",
          "value": "Köln"
        },
        "country": {
          "status": "match",
          "value": "DEU"
        }
      }
    },
    "identificationDocument": {
      "type": "1",
      "number": "T78154711V4",
      "dateIssued": "2011-05-20",
      "dateOfExpiry": "2021-05-19",
      "authority": "Landeshauptstadt Berlin",
      "placeOfIssue": "Berlin",
      "countryOfDocument": "DEU"
    },
    "signedDocuments": [
      1
    ]
  }
]
}
```

6.4 Errors

HTTP status codes in the response for error situations:

Http Status Code	**Message**	**Possible Cause**
401	Unauthorized	Wrong or missing credentials, e.g. - Wrong or missing username or password
403	Forbidden	Missing authorization, e.g. • user not authorized for the endpoint • user not authorized for the clientId

Http Status Code	Message	Possible Cause
404	Not Found	• caseId was not found

6.5 Encryption

Additionally to the encryption of HTTPS the result data is asymmetrically encrypted with a public key provided by you. The key is an additional parameter in the HTTP header of the GET requests. The cipher is transmitted in JWE format. You can decrypt the received data with your private key.

> **Unencrypted Result Data in Test Environment**
>
> During the integration of the SCR API the encryption can be configured as optional. So the HTTP header fields "x-scr-key" and "x-scr-keyhash" can be omitted in your request. The response will not be encrypted.
> If the headers are sent, the result will be encrypted.
> In the productive environment the encryption is mandatory. It will be activated after a successful encryption test.

The encryption mechanism is identical to the one used by POSTIDENT SCR-Ident API. So for details, please relate to SCR-Ident API Guide 3 - Encryption.

7 REST API - Retrieve detailed ident results via POSTIDENT SCR-Ident API

The signing process bases on a identification that is processed by the regular POSTIDENT system, this could be an identification via Videochat or with Online ID function. To query extended information related to the identification part of the process, like the pictures of the identification documents (if you are a GwG client) you have to use the POSTIDENT SCR-Ident API . To use the SCR-Ident API you can use the same credentials as for the SCR-Signing API. Each singer has a related identification case, whose case ID is returned in the field "identCaseId", when retrieving a signing case result.

The related identification case has the signing case ID as reference ID. The identity data and custum data of your signing case are also used in the identification case and will be delivered in the ident result.

8 Webhook

When this feature is activated for your account, you can receive a push notification when the idendification of a signer was successfull, all signer signed the documents or the signing case has been declined, see also the overview of Statuses of a Signing Case and a Signer.

For this purpose the POSTIDENT E-Signing system sends a POST request (from IPs 165.72.200.13, 199.40.127.49 or 156.137.9.65) to the webHookURL parameter provided by your initial call to start the signing case. Please note that the webHookURL must not contain GET parameters (e.g. https://foo.bar.com/postident/callback?ref=123) since the REST standard disallows mixing GET and POST parameters. If desired, HTTP Basic Authentication can be configured for your client id. Self-signed TLS certificates are not supported.

The POST request contains application/json with the signing case ID and an optional referenceId:

```
{
 "caseId": "<caseId>",
 "referenceId": "<referenceId>"
}
```

You can use this notification to automatically trigger a GET request with the REST API to retrieve the current result of this signing case ID.

The POSTIDENT E-Signing system expects http-status code 200 within 2 seconds, otherwise the POSTIDENT E-Signing system retries this process up to 3 times. 2 seconds is the maximum amount of time we can wait for the answer. During the waiting time resources of our system are blocked and so a longer timeout would be a risk to the stability of our system, due to the large amount of webhooks we must send to our clients. Our recommendation is, to process the webhook asynchronously, that is to send the response to the webhook immediately and to decouple the actual processing from this. This could be done by running the processing in a separate process/thread or by buffering a job in a message queue.

If activated in the client configuration portal (FA portal) the webhook data contains the referenceId that was given on the REST call when created the signing case.

ⓘ **Note**

If you want to use this feature, your domain has to be added to the whitelist in our communications infrastructure. This takes typically 7 working days. You can vary the url after the domain as you like.

9 Availability Check of POSTIDENT E-Signing system

SCR-Signing provides a particular 'alive' resource to check the availability of the POSTIDENT E-Signing system. Protocol, security and header for the alive resource are the same as above. The alive resource can be accessed under the same authentication, which is used for the SCR-Signing service, or with a dedicated alive service account. The dedicated account can only access the alive resource. Access to case data is denied, so it can be used by monitoring systems that run in another environment.

9.1 Path

Check the current availability of the POSTIDENT E-Signing system by using the following URI:

```
    /api/scr-signing/{version}/{clientId}/alive
```

The URI contains the following elements:

Element	Description	Example
version	Use „v2“	v2
clientid	Provided by Deutsche Post. Uniquely identifying your access to the API. Format: alphanumeric, uppercase (case sensitive).	1234ABCD

Example:

```
GET /api/scr-signing/v1/1234ABCD/alive
```

9.2 Response

If available, SCR-Signing will respond with the following JSON object:

```
{
  "status": "OK"
}
```

The JSON object contains only one field:

Parameter	Mandatory	Max. Length	Description	Example
status	no	3	Delivers the result of the alive check. Two values are possible: • OK • NOK	OK

There are three scenarios which are possible in this context:

1. The system is available and running fine
2. The first stage is running fine, but at least one component of the system is not available
3. The whole system is not available

SCR-Signing alive service will response differently in all above mentioned cases:

1. The system delivers an "OK"
2. The system delivers a "NOK"
3. The system delivers a HTTP error

ⓘ **Note**

Please do not call this service more than 2 times per minute.

9.3 Errors

Http Status Code	Message	Possible Cause
401	Unauthorized	Wrong or missing authorization key, e.g. - Wrong or missing username or password - clientId not found or not configured for usage of Standard Connect API
500	Internal server error	If the system has an internal issue, status code 500 will be delivered together with an error message.

10 REST API - JSON structures

This chapters describes the JSON request and response objects of the SCR-Signing Endpoints.

⚠ Please be aware that it is possible that new fields will be added to the data model in the future. Make sure that your implementation can handle unknown fields in the result data.

Newly added fields will not lead to a new version of the SCR API.

    - customData
    - identificationDocument
    - resultValue
    - addressResult
- Status Response Data
    - accountingDataStatus
    - signerDataStatus
    - identityDataResultStatus
    - statusResultValue

10.1 Start a Signing Case

10.1.1 Input Data

orderDataSigning

Your configuration and initial data to start a signing case. Order data consists of four properties:

Parameter	**Mandatory**	**Type**	**Description**
customData	no	customData	Custom properties which will be send back along with the result data
processData	yes	processData	Properties which control the behavior, such as callback URLs.

Parameter	Mandatory	Type	Description
documents	yes	document Array	Documents to be signed. All documents have to be marked with hasToBeSigned = true. A signing case is restricted to a maximum of 5 documents to be processed. The upper limit for the total size of all pdf documents belonging to the same signing case is 15 MB.
signers	yes	signer Array	Contains the data of the signers. There must be at least one signer. Maximum number of signers is 5.

customData

Parameter	Mandatory	Max. Length	Type	Description	Example
custom1	no	100	String	Custom text field in order to pass your own identifiers, labels etc. You will get this information back in the result data.	"Kunden-Nr: 1234234"
custom2	no	100	String	See custom1	
custom3	no	100	String	See custom1	

Parameter	Mandatory	Max. Length	Type	Description	Example
custom4	no	100	String	See custom1	
custom5	no	100	String	See custom1	

processData

Parameter	Mandatory	Max. Length	Type	Description	Example
caseName	yes	150	String	The display name of the case to be shown to the user in the signing process	"Kreditantrag"
targetCountry	no	3	String	Country from which the identification is requested. ISO 3166-1 ALPHA-3. If not provided, E-Signing application will set the default value DEU.	"DEU"
preferredLanguage	no	5	String	Preferred language of the user. Possible values: DE_DE, EN_UK. If not provided, E-Signing application will set the default value DE_DE. NOTE: EN_UK not yet supported. = ['DE_DE', 'EN_UK']	"DE_DE"

Parameter	Mandatory	Max. Length	Type	Description	Example
webHookUrl	no	500	String	URL for push notifications back to your application. The URL will be called via POST request. Parameter is application/json with the body {"caseId"="##caseId##"}. Only secure HTTPS URLs are supported.	"https://democompany.com/api/piwebhook"
referenceId	no	14	String	The reference id of the client. If provided, this must be unique in the context of the client id	"K2345ASDF"
validUntil	yes	10	String	The signing case URL will expire and the signing case will be declined at the end of this day in case the documents aren't signed yet. ISO 8601 format Maximun 30 days	"2016-01-28"
phoneNumberClientCustomerService	no	20	String	Will be shown to the user during the signing process. The phone number can start with "+" and contain 0-9 and blanks	"0228 12 34 56 78"
callbackUrlSigningDeclined	no		multiPlatformUrl	Redirect or callback URL for user if signing case is declined	

Parameter	Mandatory	Max. Length	Type	Description	Example
userMustConfirmHavingReadDocuments	no		Boolean	Mark "true" if user has to confirm the reading of the documents before signing the documents	false, true
noteClientCustomerService	no	100	String	A note regarding the client customer service. Will be shown below the client customer service phone number in the frontend.	
signingButtonLabel	no		Integer	Key for the text that will be displayed on the button to sign the documents. Possible values: 1- 5 1. Rechtsgültig unterschreiben 2. Zahlungspflichtigen Vertrag unterschreiben 3. Digital unterschreiben 4. Unterschreiben 5. Kostenpflichtig kaufen	1

multiPlatformUrl

Parameter	Max. Length	Type	Description	Example
webUrl	200	String	URL for web browser	"https://musterbank.eu/pi-signing-callback/success.html"

document

Parameter	Mandatory	Max. Length	Type	Description	Example
name	no	50	String	The display name of the document to be shown to the user in the signing process	"Kreditvertrag"
referenceId	no	30	String	External ID of the client for the document	"1243hiu023"
hasToBeSigned	yes		Boolean	Mark with true if document has to be signed. All documents have to be marked with hasToBeSigned = true.	true
mimeType	yes	35	String	Mime type of the document. = ['application/pdf'],	"application/pdf"

Parameter	Mandatory	Max. Length	Type	Description	Example
fileName	yes	254	String	Must end with the file type .pdf; allowed characters: a-z A-Z 0-9 _	"mydocument.pdf"
documentData	yes	5 MB	String	base64 binary document data	"TWFuIG...VyZS4="
signatureFieldName	no	100	String	Name of the pdf form field that should be used for holding the signature stamp. The field must be of type "signature" and have the size 72,5 mm x 24,0 mm. This field can be overwritten by the signatureFieldName in SignerDocumentInfo. This parameter takes precedence over signatureStampPosition.	
signatureStampPosition	no		signatureStampPosition	Position of the signature stamp. This Field could be overwritten by the signatureStamp Position in SignerDocumentInfo.	

signer

the data of the signer

Parameter	Mandatory	Type	Description	Example
contactData	yes	contactData	Contact data of the person to sign the documents (user)	
identityData	yes	identityData	Identity data of the person to sign the documents	
callbackUrlSigningSuccess	no	multiPlatformUrl	Redirect or callback URL for user after success signing	
documentsToSign	no	signersDocumentInfo Array	Contains signer related data for the documents he has to sign. Using this parameter you can specify which documents the signer has to sign and the position of the signature of the stamp for this signer for each document. If you provide signerDocumentInfo Array this signer will see, sign and download only documents in this array. In this case each document has to be signed by at least one signer. If you want the signature stamp to be placed in only in one of your documents you have to specify which documents must be signed by this signer without the signature stamp. If you don't provide this parameter all signers will sign all documents and use the configuration provided in document (field). If in document no parameter are defined also, there will be no signature stamp on the pdff documents.	

signersDocumentInfo

Parameter	Mandatory	Max. Length	Type	Description	Example
documentToSign	no	50	Integer	Index of the document to be signed by signer. The index is a pointer in the documents list of the case. First document in the list has the position 1.	1
signatureFieldName	no	100	String	Name of the pdf form field that should be used for holding the signature stamp. The field must be of type "signature" and have the size 72,5 mm x 24,0 mm. This parameter takes precedence over signatureStampPosition	
signatureStampPosition	no		signatureStampPosition	Position of the signature stamp	

signatureStampPosition

The size of the signature stamp is 72,5 mm x 24,0 mm. The signature stamp expands from the top left corner.

Parameter	Mandatory	Max. Length	Type	Description	Example
pageNumber	no	10	Integer	the signature page index	4

Parameter	Mandatory	Max. Length	Type	Description	Example
left	no	10	Integer	the x-coordinate of left edge of signature image (in units of 1/72 inches). Values are specified as offset from left margin of page	380
top	no	10	Integer	the y-coordinate of top edge of signature image (in units of 1/72 inches). Values are specified as offset from top margin of page	10

contactData

Parameter	Mandatory	Max. Length	Type	Description	Example
mobilePhone	yes	20	String	Country code plus phone number. ⚠ Must be unique in a signing case. Blanks are allowed and will be ignored. Must start with a country code beginning with "+".	- German number: "+49171123456789" or "+49 171 123456789" - US number: "+1123456789010" or "+1 123 456789010"
email	yes	320	String	⚠ Must be unique in a signing case.	"maria.mustermann@musterdomain.de"
epost	no	320	String		

identityData

Parameter	Mandatory	Max. Length	Type	Description	Example
title	no	35	String		"Dr."
firstName	yes	35	String	All first names of the person to sign the documents	"Maria"
lastName	yes	35	String	Exact last name; may include title like "Dr."	"Musterfrau"
birthName	no	35	String	Only if differing from last name. Do not include prefixes like „geb." or „Geborene"	"Rossi"
birthDate	no	10	String	ISO 8601 format: YYYY-MM-DD	"1985-01-01"
birthPlace	no	55	String		"Berlin"
nationality	no	3	String	two-letter ISO3166-1 alpha-2, three-letter ISO3166-1 alpha-3) and RKS / XK for Kosovar	"DEU"
address	no		address		

Parameter	Mandatory	Max. Length	Type	Description	Example
identityVerified	no		Boolean	Flag to indicate if the provided data has already been verified by the client. If true the identification will be skipped. Requires a corresponding contractual agreement and special setup of client configuration.	false, true

address

Parameter	Mandatory	Max. Length	Type	Description	Example
streetAddress	no	55	String	May include street name, house number, Post Office Box	"Musterstraße 2"
appendix	no	55	String		"Am Vorderhaus"
postalCode	no	11	String		"51456"
city	no	55	String		"Berlin"
country	no	3	String	ISO-3166 ALPHA-3 plus RKS for Kosovo	"DEU"

10.1.2 Response

You will get as response:

Parameter	Type	Description	Example
caseId	String	Unique Id for your signing case	"M3FB00URX4A3"
signers	signer		

signer

Parameter	Type	Description	Example
signerNumber	String	Number of the signer	"1"
webStart	webStart		

webStart

Parameter	Type	Description	Example
caseURL	String	URL to redirect the browser of the user from your web portal to the E-Signing portal. The redirection must be executed as a HTTP Post that contains a JSON Web Token ⚠ This URL is personalized for the user to be identified, please treat as confidential and share only with the respective user.	`"https://postident.deutschepost.de/signingportal/entry/e29c1298-5bff-4b0e-aa53-30b93d533840"`

Parameter	Type	Description	Example
redirectTokenSecret	String	Base64 encoded 256-Bit secret for signing the JSON Web Token	`"icEfRPW4exlKe0nDsXSHoyk7uQpupdFaFwWyT1Z8Ub8="`
resumeCaseURL	String	URL to let the user asynchronously resume the signing case for example after an interruption or waiting time. Send this URL via Email to your user to resume the signing case. ⚠️ This URL is personalized for the user to be identified, please treat as confidential and share only with the respective user.	`"https://postident.deutschepost.de/signingportal/reentry/e29c1298-5bff-4b0e-aa53-30b93d533840"`

10.2 SigningCaseResult

The result data of the signing case. SigningCaseResult consists of the caseId and the following properties:

Parameter	Cardinality	Max. Length	Type	Description	Example
caseId	1	12	String	Unique Id for your signing case	"M3FB00URX4A3"

Parameter	Cardinality	Max. Length	Type	Description	Example
caseStatus	1		String	Possible values: • new • signing process • signed • closed • declined	"signed"
caseSubStatus	1		String	Possible values: • for caseStatus declined: • validity period expired • signer was declined • for caseStatus closed: archived	"validity period expired"
created	1	26	String	Creation time of the signing case	"2017-01-28T23:59:59+01:00"
signedDocuments	0..1		signedDocuments		
accountingData	0..1		accountingData		
customData			customData		

Parameter	Cardinality	Max. Length	Type	Description	Example
referenceId	0..1	30	String	Your reference of the signing case	"2345asfd12"
signers	1..*		signerResult		
validUntil	0..1	10	String	The signing case URL will expire and the signing case will be cancelled at the end of this day. ISO 8601 format	"2017-01-28"
storeUntil	0..1	10	String	The signed documents will be available for download by the user and via SCR-Signing until the end of this day. Afterwards the signing case will be transitioned to the state closed. ISO 8601 format	"2017-01-28"

10.2.1 signerResult

Parameter	Cardinality	Max. Length	Type	Description	Example
signerNumber	1		String	Number of the signer	"1"
identCaseId	0..1	12	String	Id of the corresponding ident case, available beginning with status identification. Can be used to query ident status and detailed ident data from the SCR-Ident API	"A3KF00URX2A9"

Parameter	Cardinality	Max. Length	Type	Description	Example
created	1	26	String	Creation time of the signer	"2017-01-28T23:59:59+01:00"
modified	0..1	26	String	Last modification time of the signer	"2017-01-28T23:59:59+01:00"
identificationTime	0..1	26	String	Date and time of successful identification. ISO 8601 format, accuracy in seconds, the offset to Zulu time ±hh:mm at the end	"2017-01-28T23:59:59+01:00"
identificationMethod	0..1	35	String	Method used for identifying the signer: 'video', 'eid'	"video"
signingTime	0..1	26	String	Date and time of successful signing of the documents. ISO 8601 format, accuracy in seconds, the offset to Zulu time ±hh:mm at the end	"2017-01-28T23:59:59+01:00"
termsAndConditionsAcceptedTime	0..1	26	String	Time when signer accepted E-Signing terms and conditions	"2017-01-28T23:59:59+01:00"

Parameter	Cardinality	Max. Length	Type	Description	Example
notifyUserAboutSigningState	0..1		Boolean	Only relevant if E-Mail communication by E-Signing System is deactivated: If true: ▪ if identification succeeded you have to notify the user to continue the signature process (signer state equals "signing process") ▪ if identification failed you have to notify the user that signature process is declined (signer state equals "declined" and subStatus equals "ident declined").	false, true
userReadDocumentsTime	0..1	26	String	Time when signer confirms reading all documents. Will be provided if this feature was specified in the signing case using SCR-Signing POST	"2017-01-28T23:59:59+01:00"
signerStatus	1	100	String	Status of the signing case 'new', 'identification', 'signing process', 'signed', 'finished', 'declined'	"identification"

Parameter	Cardinality	Max. Length	Type	Description	Example
signerSubStatus	0..1	255	String	Substatus describing the status on a more detailed level. Possible values: signerStatus 'declined': • signing declined • mobil phone number not verified • ident declined • maximum number of started operations exceeded • maximum number of created certificates exceeded • maximum number of sent sms exceeded • other signer was declined signerStatus 'declined', 'signing process' • prolongued duration of identification postprocess	"mobil phone number not verified"
signerContactData	0..1		signerContactData		
identityData	0..1		identityDataResult		
identificationDocument	0..1		identificationDocument		

Parameter	Cardinality	Max. Length	Type	Description	Example
signedDocuments	1		Integer Array	Indexes of the documents signed by the given signer. The indexes are pointers in the documents list of the case. First document in the list has the position 1.	[1,2]

10.2.2 signedDocuments

Parameter	Cardinality	Max. Length	Description	Example
referenceId	0..1	30		
mimeType	1	35		"application/pdf"
fileName	1	254	Must end with the file type .pdf; no special characters allowed	"mydocument.pdf"
documentData	1		base64 binary document data	"ABFuIG...VkPcD3="

10.2.3 signerContactData

Parameter	Cardinality	Max. Length of "value" attribute	Type	Description	Example
mobilePhone	1	20	resultValue	of the user	
email	1	320	resultValue	of the user	
epost	0..1	320	resultValue	of the user	

10.2.4 identityDataResult

Parameter	Cardinality	Max. Length of "value" attribute	Type	Description
title	0..1	35	resultValue	
firstName	1	55	resultValue	
lastName	1	55	resultValue	
birthName	0..1	55	resultValue	
birthDate	1	10	resultValue	

Parameter	Cardinality	Max. Length of "value" attribute	Type	Description
birthPlace	1	55	resultValue	
nationality	1	3	resultValue	ISO-3166 ALPHA-3 plus RKS for Kosovo
address	0..1		addressResult	

10.2.5 accountingData

Parameter	Cardinality	Max. Length	Type	Description	Example
accountingNumber	0..1	14	String	Unique identifier for invoicing (dt. Abrechnungsnummer); also used as identifier of a client configuration	"37051234567891"
accountingProduct	0..1	10	String	Product displayed on invoice, e.g. E-Signing	"906800034"

10.2.6 customData

Parameter	Mandatory	Max. Length	Type	Description	Example
custom1	no	100	String	Custom text field in order from orderData	"Kunden-Nr: 1234234"
custom2	no	100	String	See custom1	
custom3	no	100	String	See custom1	
custom4	no	100	String	See custom1	
custom5	no	100	String	See custom1	

10.2.7 identificationDocument

Parameter	Cardinality	Max. Length	Type	Description	Example
type	1	10	String	Type of the document. 1 = ID Card (Personalausweis), 2 = Passport (Reisepass), 3 = Residence Title (Aufenthaltstitel), 4 = Temporary ID Card (Vorläufig ausgestellter Personalausweis), 5=Temporary Passport (Vorläufig ausgestellter Reisepass), 6 = 1954 Convention Travel Document (Stateless Person) (Reiseausweis für Staatenlose (Übereinkommen von 1954)), 7 = 1951 Convention Travel Document (Refugee) (Reiseausweis für Flüchtlinge (Übereinkommen von 1951)), 8 = 1946 Convention Travel Document (Foreigner) (Reiseausweis für Ausländer (Abkommen von 1946)), 9 = Service Passport (Dienstpass), 10 = Diplomatic Passport (Diplomatenpass), 11 = Official Passport (Ministerialpass), 12 = Official or Diplomatic Passport (Ministerial- oder Diplomatenpass)	"1"
number	1	20	String		""
dateIssued	0..1	10	String	ISO 8601 format: YYYY-MM-DD	"2002-02-01"
dateOfExpiry	1	10	String	ISO 8601 format: YYYY-MM-DD	"2021-02-09"
authority	0..1	100	String		"Berlin"
placeOfIssue	0..1	55	String		"Berlin"

Parameter	Cardinality	Max. Length	Type	Description	Example
countryOfDocument	1	3	String	ISO-3166 ALPHA-3 plus RKS for Kosovo	"DEU"

10.2.8 resultValue

Parameter	Mandatory	Max. Length	Type	Description	Example
status	no	50	String	Possible outcomes: • unchecked: value has not been verified during the identification process. This happens for example if the identification document doesn't comprise a street address. In this case the value is returned which was passed in on case creation. • new: value was not provided by client • match: value as provided by client • change: value was modified during identification process	"match"
value	no		String		"+49171123456789"

10.2.9 addressResult

Parameter	**Cardinality**	**Max. Length of "value" attribute**	**Type**	**Description**
streetAddress	0..1	55	resultValue	
appendix	0..1	55	resultValue	
postalCode	0..1	11	resultValue	
city	0..1	55	resultValue	
country	0..1	3	resultValue	ISO-3166 ALPHA-3 plus RKS for Kosovo

10.3 Status Response Data

Status response data consists of the following properties:

caseId	1	String		Unique Id for your signing case	"M3FB00URX4A3"
caseStatus	1	String		Possible values: - new - signing process - signed - closed - declined	"signed"
created	1	String	26	Creation time of the signing case	"2017-01-28T23:59:59+01:00"
accountingData	1	accountingDataStatus			
referenceId	1	String	14	Your reference of the signing case	"K2345ASDF"
signers	1..*	signerDataStatus Array			
validUntil	1	String	10	The signing case URL will expire and the signing case will be cancelled at the end of this day. ISO 8601 format	"2016-01-28"
storeUntil	0..1	10	String	The signed documents will be available for download by the user and via SCR-Signing until the end of this day. Afterwards the signing case will be transitioned to the state closed. ISO 8601 format	"2017-01-28"
Parameter	**Cardinality**	**Type**	**Max. Length**	**Description**	**Example**

10.3.1 accountingDataStatus

accountingNumber	0..1	String	14	Unique identifier for invoicing (dt. Abrechnungsnummer); also used as identifier of a client configuration	
accountingProduct	0..1	10	String	Product displayed on invoice, e.g. E-Signing	"906800034"
Parameter	**Cardinality**	**Type**	**Max. Length**	**Description**	**Example**

10.3.2 signerDataStatus

signerNumber	1		String	Number of the signer	"1"
signerStatus	1	100	String	Status of the signing case 'new', 'identification', 'signing process', 'signed', 'finished', 'declined'	"identification"
identCaseId	0..1	String	12	Id of the corresponding ident case, available beginning with status identification. Can be used to query ident status and detailed ident data from the SCR-Ident API	"A3KF00URX2A9"
created	1	String	26	Creation time of the signer	"2017-01-28T23:59:59+01:00"
modified	0..1	26	String	Last modification time of the signer	"2017-01-28T23:59:59+01:00"
identificationTime	0..1	26	String	Date and time of successful identification. ISO 8601 format, accuracy in seconds, the offset to Zulu time ±hh:mm at the end	"2017-01-28T23:59:59+01:00"
identificationMethod	0..1	35	String	Method used for identifying the signer: 'video', 'eid'	"video"
signingTime	0..1	String	26	Date and time of successful signing of the documents. ISO 8601 format, accuracy in seconds, the offset to Zulu time ±hh:mm at the end	"2017-01-28T23:59:59+01:00"
identityData	1	identityDataResultStatus		Always return the data that was given by the partner with creation of the signing case. It does not return the possibly changed identity data after the identification.	
Parameter	**Cardinality**	**Type**	**Max. Length**	**Description**	**Example**

10.3.3 identityDataResultStatus

Parameter	Cardinality	Max. Length of "value" attribute	Type	Description
firstName	1	55	statusResultValue	
lastName	1	55	statusResultValue	

10.3.4 statusResultValue

Parameter	Mandatory	Max. Length	Type	Description	Example
value	no		String		"+49171123456789"

11 PDF document requirements

This guide describes the requirements on the PDF documents to be signed by POSTIDENT E-Signing.

11.1 Format support for input documents

The pdf documents should be compliant to the PDF/A-2b format to ensure best compatibility and interoperability.

11.1.1 Conversion of alternative formats

In exceptional cases these other formats can be used, which get converted to PDF/A-2b on the POST of the signing case:

- PDF/A-1
- PDF 1.4
- PDF 1.5
- PDF 1.6
- PDF 1.7

If these formats are used, the documents must at least comply with these requirements:

- No embedded audio or video content
- No Javascript or other executable content
- All used fonts must be embedded in the document
- No use of encryption

11.2 Preliminary checks

You can check the compliance of the documents to PDF/A-2b in advance with the open source software veraPDF or the commercial Adobe Acrobat Pro. Additionally Deutsche Post offers a conversion function in its "Signing Testapp"

⚠ If, due to the use of the alternative formats, a conversion is required, this can result in deviation of the visual represenation of the original and converted document. For that reason, test documents should be converted in advance and checked for visual differences.

11.2.1 File count restrictions

A signing case is restricted to a maximum of 5 documents to be processed.

11.2.2 File size restrictions

The hard upper limit for the size of a single pdf document is 5MB.

The upper limit for the total size of all pdf documents belonging to the same signing case is 15MB.

It is recommended to reduce the file size (for example by using optimization or shrinking tools) as much as possible, since this provides a faster processing and better user experience.

11.2.3 Further format restrictions and recommendations

Form Fields

The documents should not contain PDF Form fields, because PDF Viewers can display confusing messages if such a document is signed.

Colorspace recommendations

Please use only images and graphics with a RGB colorspace for best visual representation and format compliance.

11.3 Format of the output documents

The signed documents always comply with PDF/A-2b.

12 Testing SCR-Signing API using E-Signing Test-APP

We provide a E-Signing Test-App to simulate identification and signing status to test your integration of the SCR-Signing API on our test environment.

Example test flow:

1. **Start a signing case** using SCR-Signing POST
2. **Identification:** you can simulate the identificaion with this Test-App:
    a. Find your signing case in the Test-App by signing case Id or client ID and time period
    b. Transform the state of the signer to the state “identification”
    c. Now you can test successful or declined identification status. In both cases a webhook will be sent if this feature is activated for your account
3. **Signing documents:** With a successful identification you can test successfull or declined signing cases. In both cases a webhook will be sent if this feature is activated for your account
    a. You can simulate the signing process in the Test-App or sign the documents in the Signing-Portal. If you simulate the signing process in the Test-App, the documents will not be signed. Just the state of signing case will be transformed to “signed”. You will get unmodified documents as result of the signing process
    b. To proceed the signing process in the Signing-Portal you have to use the resume URL and authorize yourself with the SMS-TAN. Then you can request the TAN for signing und confirm it, in this case a test signature will be applied in the documents
4. **Retrieve signing result** with SCR-Signing GET

If you use the E-Signing Test-App you won't get results via SCR-Ident API because the identification is mocked in this case.

13 Signing Process with a Partner

The E-Signing system allows the client to outsource the signing frontend process to third parties (partners), e.g. a comparison portal. This partner can take over the initialization of the signing case and the communication with the private customer for the client. The client only needs to retrieve the signed PDF documents via SCR and, if necessary, SFTP. This chapter describes the special features for this client - partner constellation.

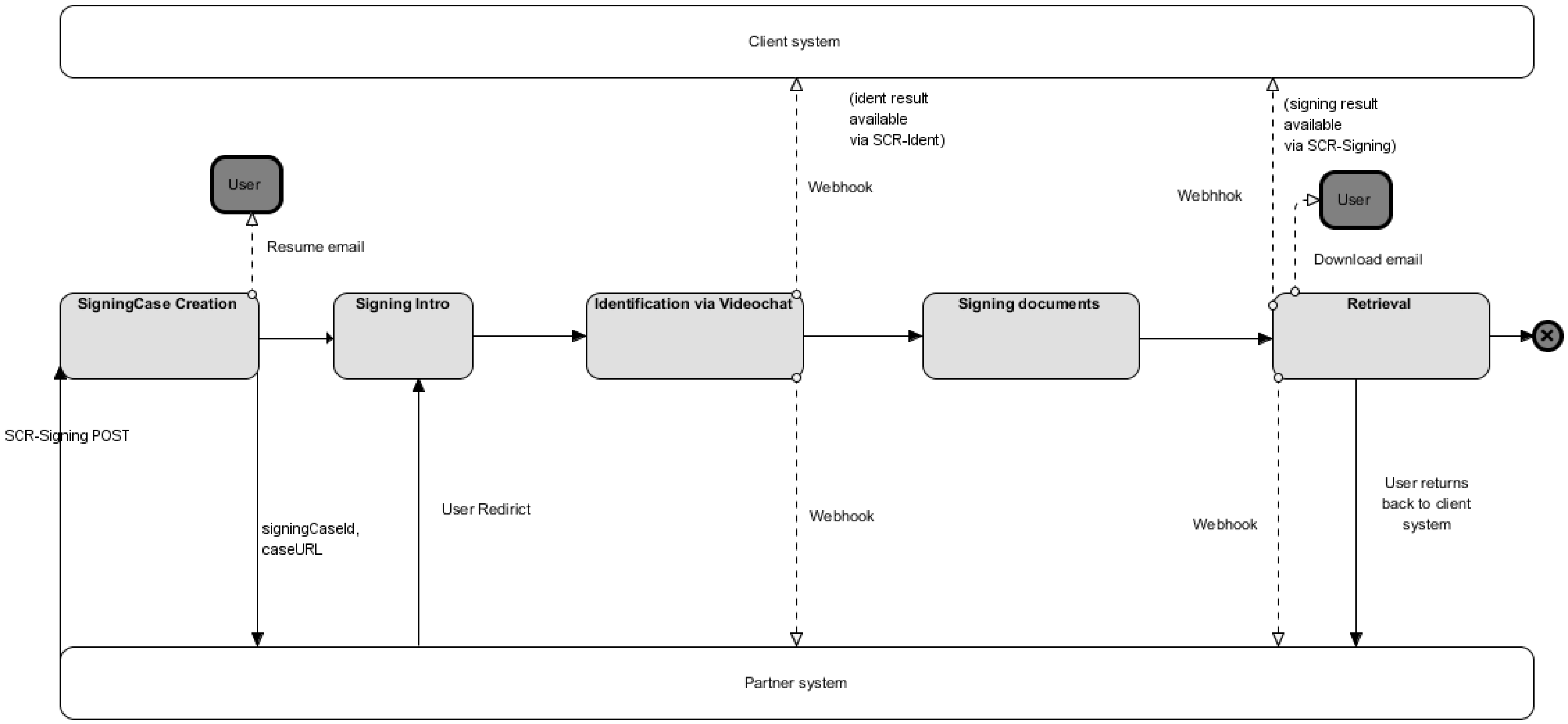
In order to carry out the process in the frontend, the partner has the following options:

- Start a signing process for the client via SCR-Signing
- Receive webhooks when the status of the signing case changes
- Retrieve status information about the operation via SCR-Signing, e.g. to take over the communication with the private customer.

The partner, however, can not query result data of the signing case, but only its status.

In order to make the origin of the transactions comprehensible, only the partner can create the signing case in this constellation. To represent this client - partner constellation, a separate accounting number will be used. The client can query the signed PDF documents as usual and can also be informed about status changes of the signing case via Webhook - there are no deviations from the standard process.

13.1 General Flow for one Signer



13.2 Start a new signing case

The partner can create a new signing case for the client via SCR-Signign POST

```
/api/scr-signing/{version}/{clientId}/signingcases
```

13.3 Receive webhooks

The partner can receive webhooks when the status of an signing case changes. For this he needs to pass in the POST parameters webHookUrlPartner its URL. Only secure HTTPS URLs are supported.

```
{
 ...
 "processData": {
    ...
    "webHookUrlPartner": "https://webhook.partner.com",
    ...
  },
  ...
}
```

13.4 Retrieve status information

The partner can retrieve the status information about the signing case via SCR-Signing after being informed about the status change via Webhook.

13.4.1 Path

Request the current status of a signing case.

```
GET /api/scr-signing/{version}/{clientId}/signingcases/{caseId}/status
```

13.4.2 Sample Request and Response

Request to retrieve a single signing case status by case ID:

```
GET /api/scr-signing/v2/1234ABCD/signingcases/YR9W91GEZK24/status HTTP/1.1
Host: postident.deutschepost.de
Authorization: Basic R0s0Mi5TQ1I6cEpzZW43NWh3biF0
```

Response for successful signing case status:

```
{
"caseId":"SCRTEST75",
 "caseStatus":"SIGNED",
 "created":"2018-02-27",
 "accountingData":{
     "accountingNumber":"ACC"
  },
 "referenceId":"15701",
 "signers":[
    {
     "signerNumber": "1",
     "identCaseId":"FHD2TSEAN4ZX",
     "created":"2017-02-27",
     "modified": "2020-03-25",
     "identificationTime": "2020-03-25T13:14:50+01:00",
     "identificationMethod": "video",
     "signerStatus": "signing process",
     "signingTime":"2018-03-13T11:56:30+01:00",
     "identityData":{
         "firstName":{
             "value":"Maria"
         },
         "lastName":{
             "value":"Musterfrau"
         }
      }
    }],
  "validUntil":"2018-03-15"
}
```