CS REST API 1.0

REST API for
Ciceron Certificate Server

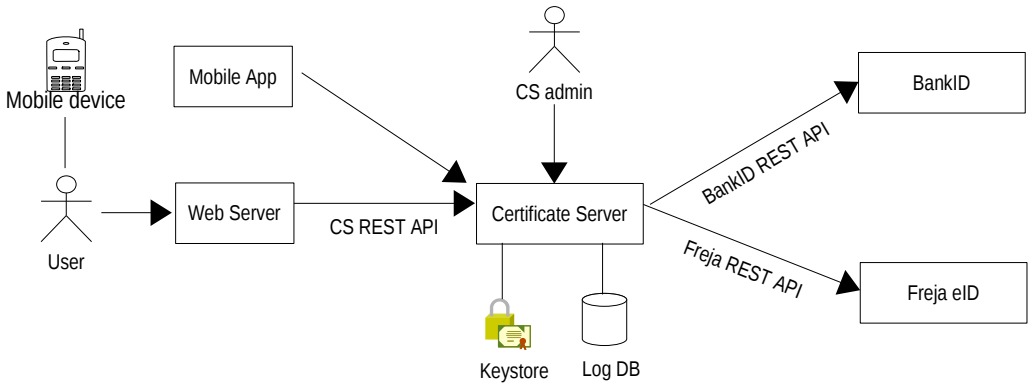
2025-05-08	Ver 5	Accept initial calls without a personalNumber (for native apps or QR code handling)
2024-09-03	Ver 4	Add information about callInitiator
2023-09-14	Ver 3	twoday rebranding, added information about the new BankID API
2022-09-21	Ver 2	Cancel method added
2020-09-09	Ver 1	First release

Table of Contents

Introduction

This document describes CS REST API which is used to communicate with *Ciceron Certificate Server (CS*) when performing authentication operations using BankID or Freja eID+.

Overview using CS REST API



BankID API changes

The BankID API version 5.1 has an end-of-life in may 2025. This API is not changed because of this, but the Certificate Server has been changed so it can route the calls using the new endpoint and its new API description (using the /phone/auth interface so that a personal number can still be used).

Authentication

The URL **/rest/auth** is called using HTTPS POST with a Content-Type header containing **application/x-www-form-url-encoded** or **multipart/form-data**. The body contains the form values read by CS. The returned response contains Content-Type **application/json** and the response body will contain the JSON result. If the HTTP result is 200 OK, the possible returned “status” codes is one of “pending” or “failed” when initializing an authentication.

Initializing an authentication (using curl)

```
curl -v -i -F system=test_system_1 -F provider=freja -F personalNumber=<valid pnr> https://<server
name>/rest/auth
> POST /rest/auth HTTP/1.1
> Host: <server name>
> User-Agent: curl/7.58.0
> Accept: */*
> Content-Length: 532
> Content-Type: multipart/form-data; boundary=------------------------d8673435bdf38713
  ...
< HTTP/1.1 200 OK
< Server: Certificate Server
< Connection: keep-alive
< Content-Type: application/json; charset=utf-8
< Content-Length: 134
  ...
{"infoCode":"outstandingTransaction","orderRef":"fArPmBWnwPPlDOVEVJyNnc_pRx4eran100cVZB380jBK9j53Vc
n0HwIZ5mS91BJj","status":"pending"}
```

Description

The form values “system” and “provider” are mandatory.

- **system** a system identifier configured and ordered from twoday.
- **provider** must be “bankid” or “freja”.
- **personalNumber** (optional) a swedish personal number in the format “YYYYMMDDNNNN”.

The returned “orderRef” is later used when collecting the result of an authentication. The status “pending” is the normal response for this call and the “infoCode” will contain “outstandingTransaction”.

```
{"infoCode":"outstandingTransaction","orderRef":"fArPmBWnwPPlDOVEVJyNnc_pRx4eran100cVZB380jBK9j53Vc
n0HwIZ5mS91BJj","status":"pending"}
```

Sending the same call once again without a collect or cancel will result in an “failed” status. The returned “infoCode” will contain an error status (different for BankID and Freja) and “errorMessage” contains the description of the error.

```
{"errorMessage":"Order already in progress for
pno","infoCode":"alreadyInProgress","status":"failed"}
```

Or:

```
{"errorMessage":"Authentication request failed. Previous authentication request was rejected due to
security reasons.","infoCode":"2000","status":"failed"}
```

Starting the BankID or Freja App

This is not covered by this documentation. Please read the corresponding API documentation from BankID and Freja.
For BankID, some extra JSON parameters are also returned in the initializing call and these are:

autoStartToken – no longer used

qrStartToken – used when creating a QR code

qrStartSecret – used when creating an animated QR code

Note: There may be different starting instructions depending on Android and iOS.

Collecting authentication result

The URL **/rest/auth/collect** is called using HTTPS POST with a Content-Type header containing **application/x-www-form-url-encoded** or **multipart/form-data**. The body contains the form value “orderRef” that was returned from the initializing call. The returned response contains Content-Type **application/json** and the response body will contain the JSON result. If the HTTP result is 200 OK, the possible returned status codes are one of “pending”, “complete” or “failed”. Note: This call should be repeated (at most) every two seconds when the returned status is “pending”.

Collecting the result of an authentication (using curl)

```
curl -v -i -F orderRef=fArPmBWnwPPlDOVEVJyNnc_pRx4eran100cVZB380jBK9j53Vcn0HwIZ5mS91BJj
https://<server name>/rest/auth/collect
> POST /rest/auth/collect HTTP/1.1
> Host: <server name>
> User-Agent: curl/7.58.0
> Accept: */*
> Content-Length: 532
> Content-Type: multipart/form-data; boundary=------------------------22ba6995f45e0a9c
  ...
< HTTP/1.1 200 OK
< Server: Certificate Server
< Connection: keep-alive
< Content-Type: application/json; charset=utf-8
< Date: Wed, 09 Sep 2020 09:37:59 GMT
< Content-Length: 56
  ...
{"infoCode":"outstandingTransaction","status":"pending"}
```

Description

The returned “orderRef” from the initializing call is used when collecting the result of an authentication. If the returned status is “pending”, then the “infoCode” will be one of:

- **outstandingTransaction** The client has not yet received the order
- **noClient** The client has not yet received the order
- **started** A client has been started but a usable ID has not yet been found in the started client
- **userSign** The client has received the order

If the returned status is “complete”, the follwing JSON is returned for Freja:

```
{"email":"test.testsson@gmail.com","givenName":"Johan","personalNumber":"<valid
pnr>","status":"complete","surname":"Bj\u00F6rklund"}
```

and the follwing JSON is returned for BankID:

```
{"certNotAfter":"2022-02-02T22:59:59Z","certNotBefore":"2020-02-02T23:00:00Z","givenName":"Johan
Sebastian","personalNumber":"<valid pnr>","status":"complete","surname":"Bj\u00F6rklund”}
```

The common information returned are:

- **personalNumber**
- **givenName**
- **surname**

Canceling authentication

The URL **/rest/auth/cancel** is called using HTTPS GET or POST with a Content-Type header containing **application/x-www-form-url-encoded** or **multipart/form-data**. The query string or body contains the form value “orderRef” that was returned from the initializing call. The returned response contains Content-Type **application/json** and the response body will contain the JSON result. If the HTTP result is 200 OK, the returned status code is “cancelled”.

Using curl with POST

```
curl -v -i -F orderRef=fArPmBWnwPPlDOVEVJyNnc_pRx4eran100cVZB380jBK9j53Vcn0HwIZ5mS91BJj
https://<server name>/rest/auth/cancel
> POST /rest/auth/cancel HTTP/1.1
> Host: <server name>
> User-Agent: curl/7.58.0
> Accept: */*
> Content-Length: 532
> Content-Type: multipart/form-data; boundary=------------------------22ba6995f45e0a9c
  ...
< HTTP/1.1 200 OK
< Server: Certificate Server
< Connection: keep-alive
< Content-Type: application/json; charset=utf-8
< Date: Thu, 01 Sep 2022 10:42:01 GMT
< Content-Length: 22
  ...
{"status":"cancelled"}
```

Using curl with GET

```
curl -v -i https://<server name>/rest/auth/cancel?orderRef=fArPmBWnwPPlD91BJj
> POST /rest/auth/cancel?orderRef=fArPmBWnwPPlD91BJj HTTP/1.1
> Host: <server name>
> User-Agent: curl/7.58.0
> Accept: */*
  ...
< HTTP/1.1 200 OK
< Server: Certificate Server
< Connection: keep-alive
< Content-Type: application/json; charset=utf-8
< Date: Thu, 01 Sep 2022 10:42:01 GMT
< Content-Length: 22
  ...
{"status":"cancelled"}
```

Error handling

The normal HTTP status code is “200 OK” even if the JSON contains status “failed”. An optional “errorMessage” may also be returned in this case. Other HTTP status codes may also be returned, for instance “404 Not Found” if the case sensitive URL was misspelled. “500 Internal Error” or other gateway related errors may also be returned. Socket communication errors may also occur if the server cannot be reached or if a client certificate is invalid. Error codes returned in the “infoCode” if the status is “failed” may be different between BankID and Freja. The following error codes will be returned:

invalidParameters	Input parameters are missing or invalid. The signature has already been collected; it can only be collected once
unauthorized	The security configuration of the RP does not allow the requested operation
invalidClient	The BankID client is invalid
certificateErr	The BankID Certificate has expired or revoked
maintenance	Some kind of temporary problems. Retry later
internalError	Internal error in RpService
expired	The queried transaction has expired
alreadyInProgress	The End User already has a request to process
userCancel	Request cancelled by user
cancelled	Request cancelled
requestTimeout	The BankID App was not started within the time limit
rejected	The request was rejected. F.i if more than one transaction for the same user is run at the same time
2000	Freja Request failed. Previous authentication request was rejected due to security reasons.

Timeouts may be handled in a different way by the caller than other errors, but make sure to check both “expired” and “requestTimeout” codes.

Special handling

Default registration level of Freja is “PLUS” meaning the highest registration level for a person. The default country is “SE” so by default only a swedish person that has been verified in person by an ATG representative. This handling can be overrided with the use of the following submitted form fields:

- **country** with one of the country codes: SE (default), NO, FI, DK, GB, UA
- **minRegistrationLevel** with either EXTENDED or PLUS (default)

Depending on the country, the personalNumber must contain the personal number format of that country.

Default call initiator is “user” and BankID will then show a dialog asking if you have initiated the call. This can be changed to the opposite if the relaying party has initiated the call by submitting the following form field:

- **callInitiator** with the value “user” (default) or “RP”

If “user” is specified, the call was initiated by the citizen to the relaying party (RP).

If “RP” is specified, the call was initiated by the relaying party to the citizen.

Native app usage

To use the CS REST API with a native app, the process is described below.

• The user clicks a login button in the app.

• The app calls the rest/auth endpoint.

• When the api responds with 200 OK the app starts the provider app (BankID or Freja)
The app will be started with different “scheme” and query strings.
There can also be different handling for Android and iOS.
Please read the corresponding API documentation for BankID and Freja.

• The app polls the endpoint /rest/auth/collect while the status code is “pending”.

• When the app receives the status code “complete”, the user information will be returned.

• Take care of other status codes, for instance “cancelled”.

• The native app can continue the authenitication flow with the user information.