Public document

BORICA
BANK SERVICES

INTEGRATION GUIDE
for
The Relying parties with Cloud Qualified Electronic Signature service
provided by BORICA

Remote Identification

CONTENTS

ACRONYMS

CA	Certification Authority
CRL	Certificate Revocation List
CQES	Cloud Qualified Electronic Signature
DN	Distinguished Name
EGN	Uniform civil number assigned to each Bulgarian citizen
LNC	Uniform civil number assigned to a foreigner living in Bulgaria
eIDAS	electronic Identification, Authentication and trust Services (EU Regulation 910/2014)
EU	European Union
HSM	Hardware Security Module
OCSP	On-line Certificate Status Protocol
PIN	Personal Identification Number
PKI	Public Key Infrastructure
QC	Qualified Certificate

For additional information related to this document, please contact the Provider at:

41 “Tsar Boris III” Blvd.
1612 Sofia
BORICA AD
Tel.: 0700 199 10
E-mail: info@borica.bg
Official Web site: www.b-trust.bg

The Relying parties should integrate with Signing API WS in their signing applications (CAS) in order to send person identifiction requests. For this integration CAS SSL certificate for mutual SSL should be issued and a relying party ID should be given to CAS by B-Trust. The relying party is given access to the requested WS operations during the intregration request. All the requests from CAS to Signing API WS should be over the built mutual SSL chanel and should contain the relying party ID.

Electronic identification service is verification with subsequent validation and registration of personal data from a nationally approved identity document through video technology. It provides electronic identity to natural persons who address e-services requiring e-identification and/or authentication in the virtual environment of Relying Parties. The identification of the person is based on a created unique electronic identity including two permanent elements – an electronic identifier (eID) and an electronic identity certificate.

The Relying parties should integrate with Signing API WS in their signing applications (CAS) in order to use Electronic identification service. For this integration CAS SSL certificate for mutual SSL should be issued and a relying party ID should be given to CAS by B-Trust. The relying party is given access to the requested WS operations during the intregration request. All the requests from CAS to Signing API WS should be over the built mutual SSL chanel and should contain the relying party ID.

1 Basic Scenario 1 – Remote electronic identification of natural person with B-Trust MOBILE

1.1 Step 1: Send identification request

The Relying party’s signing application (CAS) sends identification request using /identification function (identification). CAS specifies national id, date of birth, phone number, email address and why the identity is needed. As a result of this function CAS receives callbackId with which to check the status of the requested document(asynchronous) till the end of request validity period.

URL: https://cqes-rpuat.b-trust.bg//signing-api/v2/identification
METHOD: POST
REQUEST HEADERS

KEY	VALUE	MANDATORY FIELD
Accept-language	bg || en	false
relyingPartyID	123456789	true
accept	application/json	true
Content-Type	application/json	true

REQUEST BODY

{

```
  "personIdentityData": { //Personal identity data
    "docType": "EGN", //Type of identifier(EGN, LNC, EMAIL or PHONE)
    "identificatorValue": 7805166742, //Value of identifier
    "dateOfBirth": "16-05-1978" //Birth date of the identified person
  },
  "phoneNumber": "+359888133443", //Person phone number (for onboarding)
  "email": "dnikolov@borica.bg", //Person email address (for onboarding)
  "payer": "RELYING_PARTY", //Who will be charged in order to pay for the identification operation (Relying party(RELYING_PARTY))
  "identificationReason": "TEST IDENTIFICATION REASON" //Identification reason - why the Relying party needs the identification
}
```

RESPONSE HEADERS

KEY	VALUE
Content-Type	application/json

RESPONSE BODY

Status code 200

```
{
  "data": {
    "callbackId": "a2c25bcd-00f0-4128-8890-fbcb26fc2f81", //Callback ID of the identification request. This ID is used to check the status of the identification request with /identification/{callbackId} function.
    "validity": "2021-09-14T11:25:32.132+00:00" //Response timeout - after it the request is invalid
  },
  "responseCode": "ACCEPTED", //Response code (status of the response)
  "code": "ACCEPTED", //Response code (status of the response)
  "message": "The request has been accepted." //Response message. The message can be localized with ‘Accept-language’ header
}
```

Status code 400

```
{
    "code": "BAD_REQUEST",
    "message": "The request could not be understood by the server due to malformed syntax (invalid request parameters)."
}
```

Status code 401

```
{
    "code": "UNAUTHORIZED",
    "message": "The request is unauthorized."
}
```

Status code 404

```
{
    "code": "NOT_FOUND",
    "message": "The server has not found the signed content."
}
```

Status code 500

```
{
    "code": "ERROR",
    "message": "Internal server error. The server encountered an unexpected condition which prevented it from fulfilling the request."
}
```

1.2 Step 2: CAS uses /identification/{callbackId} function (getIdentificationResult) to check the status of the requested identification (using “Polling” mechanism)

When the status becomes FINISHED the electronic identity certificate(content) and coresponding electronic identifier(identificator) can be downloaded.

URL: https://cqes-rpuat.b-trust.bg signing-api/v2/identification/{callbackId}
METHOD: GET

REQUEST HEADERS

KEY	VALUE	MANDATORY FIELD
Accept-language	bg || en	false
relyingPartyID	123456789	true
accept	application/json	true

REQUEST PARAMETERS

PATH:

- callbackId // Callback ID(request id) - result of the synchronous operation /identification request (identification)

RESPONSE HEADERS

KEY	VALUE
Content-Type	application/json

RESPONSE BODY

Status code 206

```
{
   "status": "IN_PROGRESS", //Progress of the identification request. ON_BOARDING,
IN_PROGRESS or FINISHED
   "responseCode": "OK", //Response code (status of the response)
   "code": "OK", //Response code (status of the response)
   "message": "The request has been executed successfully."//Response message. The message
can be localized with ‘Accept-language’ header
}
```

Status code 200

```
{
   "data": {
      "content":
"JVBERi0xLjUNCiW1tbW1DQoxIDAgb2JqDQo8PC9UeXBlL0NhdGFsb2cvUGFnZXMgMiAwIF...",
//eID certificate (PDF document)
      "identification_before": "BORICA", //eID identification will be used in the specified organisation
      "identification_reason": "TEST IDENTIFICATION REASON", //eID identification reason (why
the identification is needed)
```

```
      "identificator": "string" //eID unique identifier (in the relying party domain)

   },

   "responseCode": "GENERAL_OK", //Response code (status of the response)

   "status": "FINISHED", //Progress of the identification request.

   "code": "OK", //Response code (status of the response)

   "message": "Successful operation"//Response message. The message can be localized with
'Accept-language' header

}
```

Status code 400, 401, 403, 404, 500

```
{

    "data": null,

    "code": "REJECTED",

    "message": "The request is rejected."

}
```

2 Basic Scenario 2 – Remote electronic WEB identification of natural person

2.1 Step 1: CAS sends request to start identification session

CAS sends request to start identification session using /identification/web/websession/start function(startWebIdentification). CAS specifies requestCallbackUrl(to receive a message at status change) and identificationReason(why the identity is needed). As a result of this function CAS receives webSessionId(identifier of web session), resultId(identifier of the result) and webSessionExpireDate(end date of webb session validity period.

URL: https://cqes-rpuat.b-trust.bg/signing-api/v2/identification/web/websession/start
METHOD: POST
REQUEST HEADERS

KEY	VALUE	MANDATORY FIELD
Accept-language	bg || en	false
relyingPartyID	123456789	true
Content-Type	application/json	true

REQUEST BODY

```
{
  "requestCallbackUrl": "https://webhook.site/441bf812-f4dc-481b-93d6-2684b63eaacb", //at this URL the response of the identification request will be send - step 4
  "identificationReason": "TEST IDENTIFICATION" // Why the identification is needed/requested
}
```

RESPONSE HEADERS

KEY	VALUE
Content-Type	application/json

RESPONSE BODY

Status code 200

```
{
  "data": {
    "webSessionId": "245ccb3e-5105-4930-8855-b6b46b0edc3d", // web session id that would be used in next operation
    "resultId": 95, // identifier of the result
    "webSessionExpireDate": "2021-09-09T13:29:41.000+00:00" // period of validity of webSessionId
  },
  "responseCode": "ACCEPTED", //Response code (status of the response)
  "code": "ACCEPTED", //Response code (status of the response)
  "message": "Successful operation" //Response message. The message can be localized with ‘Accept-language’ header
}
```

Status code 400, 401, 404, 500

2.2 Step 2: CAS sends identification request

CAS sends identification request using /identification/web/sessions/by-otc-request/{websessionId} function(createRegistration) with the webSessionId parameter received in startWebIdentification operation. CAS specifies cancelUrl(URL where the customer will be redirected if the identification fails), deviceFingerprint(information about the customer’s device/browser), externalRef(identifier of the request in the CAS), successUrl(URL where the customer will be redirected if the identification is successful), userLanguage(bg/en), verifyEmailAddress(flag that specifies that the email should be validated in identification process), verifyPhoneNumber(flag that specifies that the phone number should be validated in identification process). As a result of this function CAS receives sessionId.

URL: https://cqes-rpuat.b-trust.bg/signing-api/v2/identification/web/sessions/by-otc-request/{websessionId}
METHOD: POST
REQUEST HEADERS

KEY	VALUE	MANDATORY FIELD
Accept-language	bg || en	false
relyingPartyID	123456789	true
Content-Type	application/json	true

REQUEST PARAMETERS

PATH:

- websessionId // Web identification web session ID - received by /identification/web/websession/start operation (startWebIdentification)

REQUEST BODY

{

"cancelUrl": "https://google.com", //at this URL the customer will be redirected if he reject identification

"deviceFingerprint": {}, // information about customer's device. Could be empty object

"externalRef": "1", // identification request identifier in the relying party system

"successUrl": "https://borica.bg", //at this URL the customer will be redirected if the identification is successful

"userLanguage": "bg", //language that will be used in identification process

"verifyEmailAddress": true, // indicates if the email address should be verified in the identification process

"verifyPhoneNumber": true, // indicates if the phone address should be verified in the identification process

"showGtcGdp": true, // default: true. If the parameter is set to false the step with general information about the process is skipped

"showMainInfo": true // default: true. If the parameter is set to false the step with GDPR information and Terms of Use is skipped

}

RESPONSE HEADERS

KEY	VALUE
Content-Type	application/json

RESPONSE BODY

Status code 200

```
{
   "userLanguage": "bg", //Language that will be used in identification process
   "verifyEmailAddress": true, //Indicates if the email address should be verified in the identification process
   "verifyPhoneNumber": true, //Indicates if the phone address should be verified in the identification process
   "successUrl": "https://borica.bg", //At this URL the customer will be redirected if the identification is successful
   "cancelUrl": "https://google.com", At this URL the customer will be redirected if he rejects identification
   "sessionId": "56f6bccb-1635-444e-b983-79bc35d22743", // Identification session identifier that will be used to construct URL link for video identification where the customer should be redirected
   "registrationId": "e9ed4675-e73e-494d-9893-602afcebe672", //Identification registration id
   "createdOn": "2021-09-09T15:49:54.407246+03:00", //Identification creation date
   "expiredOn": "2021-09-09T16:49:54.406993+03:00", //Identification expiration date
   "serviceName": "OTC_RELYING_PARTY_24102167", //Identification request service name
   "otcRequestId": "245ccb3e-5105-4930-8855-b6b46b0edc3d" //Identification otc request id
}
```

Status code 400, 401, 404, 500

2.3 Step 3: CAS redirects customer for video identification

CAS creates redirect link with sessionId - https://idtest.borica.bg/session/{sessionId}. The customer is redirected to identification page where an identification process is made (verify email, verify phone number, ID documents checks, liveness detection, selfie).

URL: https://iduat.borica.bg/session/{sessionId}
METHOD: GET

REQUEST HEADERS

REQUEST PARAMETERS

PATH:

- sessionId// this redirect link should be constructed in the relying party system. The customer is redirected to this address in order to start the video identification

2.4 Step 4: CAS receives identification response status change

CAS is notified at its requestCallbackUrl with sessionId for identification status change.

REQUEST HEADERS

KEY	VALUE
Content-Type	application/json

REQUEST BODY

```
{
   "code": "OK", //Response code (status of the response)
   "message": "The request has been executed successfully.", //Response code message (status of the response)
   "requestId": 315, //Identification request ID
   "relyingPartyId": 24102167, // Relying party ID
   "data": {
      "processState": "CREATE_SIGN_SESSION_REQUEST", // state of the identification request. Should be used in next step
      "sessionId": "245ccb3e-5105-4930-8855-b6b46b0edc3d", //session identifier of the identification response. Should be used in next step
          "resultId": "95", // identifier of the result. Should be used in next step
      "dataMessage": "The request with resultID 95 completed successfully" //Response message
   }
}
```

2.5 Step 5: CAS downloads identification response data

CAS downloads the identification response data using /v2/identification/web/{resultId}/result/{processState}/{sessionId}/{onlyMetadata} function (getWebIdentificationResults).

URL:https://cqes-rpuat.b-trust.bg//signing-api/v2/identification/web/{resultId}/result/{processState}/{sessionId}/{onlyMetadata}

METHOD: GET

REQUEST HEADERS

KEY	VALUE	MANDATORY FIELD
Accept-language	bg || en	false
relyingPartyID	123456789	true
Content-Type	application/json	true

REQUEST PARAMETERS

PATH:

- processState: CREATE_SIGN_SESSION_REQUEST // Web identification process state. Received at callback URL - step 4.
- sessionId: 245ccb3e-5105-4930-8855-b6b46b0edc3d // Web identification result ID - received by the Relying party's callback URL at status change - step 4
- resultId: 95 // result identifier. Web identification session ID - received by the Relying party's callback URL at status change – step 4
- onlyMetadata: true //true or false - indicates only metadata for create sign session will be return (response won't contain eID certificate). Only valid when it is called with processState CREATE_SIGN_SESSION_REQUEST

RESPONSE HEADERS

KEY	VALUE
Content-Type	application/json

RESPONSE BODY

Status code 200

```
{
  "data": {
    "eIDDocAsBase64": "JVBERi0xLjYKJfbk/N8KMSAwIG9iago....", // Base64 encoded PDF document with the electronic identity certificate (personal information)
    "clientIdentificator": "bf821d16f6e0dd070a6a0035e524285c78ab59d7c8d03622f43088ba0e5ea30c", // customer's identifier in the relying party domain
    "signSessionId": "491d8265-c299-41b7-b8bc-9daafdce077c", // session identifier for signing with OTC based on this web identification. Should be used in next step for signature request
```

"signSessionIdExpireDate": "2021-09-10T20:59:59.000+00:00" // validity of sign session

},

"responseCode": "OK", //Response code (status of the response)

"code": "OK", //Response code (status of the response)

"message": "The request has been executed successfully." //Response message. The message can be localized with 'Accept-language' header

}

Status code 400, 401, 404, 500

3 Basic Scenario 3 – Sign documents with OTC (one time certificate) after WEB identification

3.1 Step 1: CAS sends request sign documents wit OTC after web identification

The Relying party's signing application (CAS) sends identification request using /identification function (identification). CAS specifies national id, date of birth, phone number, email address and why the identity is needed. As a result of this function CAS receives callbackId with which to check the status of the requested document(asynchronous) till the end of request validity period.

URL: https://cqes-rpuat.b-trust.bg/signing-api/v2//identification/web/signsession/start
METHOD: POST
REQUEST HEADERS

KEY	VALUE	MANDATORY FIELD
Accept-language	bg || en	false
relyingPartyID	123456789	true
accept	application/json	true
Content-Type	application/json	true

REQUEST BODY

"identificator": "bf821d16f6e0dd070a6a0035e524285c78ab59d7c8d03622f43088ba0e5ea30c", // eID identificator value - received as result of web identification process. Received at step 5.2

"signSessionId": "43c1cd77-a2bc-42c7-9c6d-96e3dc2ebd71", // Sign session id - received as result of web identification process. Received at step 5.2

"documentsForSign": [//A list with documents that should be signed

```
    {
        "contentAsBase64": "JVBERi0xLjUNCiW1tbW1DQoxIDAgb2JqDQo8PC9UeXB...", // Base64 encoded PDF document that should be signed with OTC
        "fileName": "test.pdf", // File name of the document - will be used when the customer download the document to view the content
        "legalFileName": "TEST DOCUMENT" // Name of the document - will be used when the customer is redirected to a site where he can view the document
         “padesVisualSignature”: true// true or false - specifes if the signature in PDF signed file should be visualized in the signed file
         “signaturePosition”: { // Specifies the signature position in PDF signed file
                 “imageHeight”: 100 // Sets a height of the signature field in PDF signed document
                 “imageWidth”: 100 // Sets a width of the signature field in PDF signed document
                 “imageXAxis”: 100 // Sets a upper left X coordinate of the signature field
                 “imageYAxis”: 100 // Sets a upper left Y coordinate of the signature field
                 “pageNumber”: 1 // Sets a page number where the signature field should be placed NOTE - the counting starts from 1 (one) for the first page of the PDF document
         },
         "toBeArhieved": false // Optional parameter if relyingParty subscribed to the Qualified long-term preservation service (QLTPS)
    }
  ]
“sendSignedDocumentsByEmail”: true // Send (true) or don’t send (false) signed documents to customer
“cancelUrl”: https://google.com // At this URL the customer will be redirected if he rejects signature
“successUrl”: https://borica.bg // At this URL the customer will be redirected if the identification is successful
}
```

RESPONSE HEADERS

KEY	VALUE
Content-Type	application/json

RESPONSE BODY

Status code 200

```
{
   "data": {
      "clientDocumentUrl": "https://mytest.borica.bg/cqes/api/web-identification/a405b47c88979b6c7e91e96e2a26a89bcda4d3ebc08520247e7fffb5392afd2a",// customer must be redirected to this address in order to view and enter SMS code to sign documents
      "expireDate": "2021-09-10T20:59:59.000+00:00", // validity of the URL address
      "resultId": 109 // identifier of the result
   },
   "responseCode": "OK", //Response code (status of the response)
   "code": "OK", //Response code (status of the response)
   "message": "The request has been executed successfully." //Response message. The message can be localized with ‘Accept-language’ header
}
```

Status code 400, 401, 404, 500

3.2 Step 2: CAS receives sign response status change

CAS is notified at its requestCallbackUrl with sessionId for identification status change.

REQUEST HEADERS

KEY	VALUE
Content-Type	application/json

REQUEST BODY

```
{
   "code": "OK", //Response code (status of the response)
   "message": "The request has been executed successfully.", //Response code message (status of the response)
   "requestId": 327, //Sign request ID
   "relyingPartyId": 24102167, // Relying party ID
   "data": {
      "processState": "SIGN_SESSION", // state of the sign request. Should be used in next step
```

```
    "sessionId": "491d8265-c299-41b7-b8bc-9daafdce077c", //session identifier of the sign response. Should be used in next step

      "resultId": 109, // identifier of the result. Should be used in next step

    "dataMessage": "The request with resultID 109 completed successfully" //Response message

  }

}
```

3.3 Step 3: CAS downloads signature response data

CAS downloads the identification response data using /identification/web/{resultId}/result/{processState}/{sessionId} function (getWebIdentificationResults).

URL:https://cqes-rpuat.b-trust.bg/signing-api/v2//identification/web/{resultId}/result/{processState}/{sessionId}

METHOD: GET

REQUEST HEADERS

KEY	VALUE	MANDATORY FIELD
Accept-language	bg || en	false
relyingPartyID	123456789	successUrl
Content-Type	application/json	true

REQUEST PARAMETERS

PATH:

- processState: SIGN_SESSION // Web identification process state. Received at callback URL - step 2.
- sessionId: 245ccb3e-5105-4930-8855-b6b46b0edc3d // Signature result ID - received by the Relying party’s callback URL at status change - step 2
- resultId: 95 // result identifier. Signature result ID - received by the Relying party’s callback URL at status change – step 4

RESPONSE HEADERS

KEY	VALUE
Content-Type	application/json

RESPONSE BODY

Status code 200

```
{

  "data": {
    "signDocuments": [ // A list with signed documents
      {
        "filename": "test.pdf", // Signature file name
        "contentAsBase64": "JVBERi0xLjUNCiW1tbW1DQoxIDAgb2JqDQo8PC..." // Base64 encoded PDF document - result of the signature
      }
    ],
    "clientIdentificator": "bf821d16f6e0dd070a6a0035e524285c78ab59d7c8d03622f43088ba0e5ea30c", // customer's identifier in the relying party domain
    "signSessionId": "491d8265-c299-41b7-b8bc-9daafdce077c", // session identifier for signing with OTC based on this web identification. Should be used for next OTC signature request with the web identification
    "signSessionIdExpireDate": "2021-09-10T20:59:59.000+00:00" // validity of sign session
  },
  "responseCode": "OK", //Response code (status of the response)
  "code": "OK", //Response code (status of the response)
  "message": "The request has been executed successfully." //Response message. The message can be localized with ‘Accept-language’ header
}
```

Status code 400, 401, 404, 500

4 Basic Scenario 3 – Check the status of an registration from the registration module by registration id after WEB identification request

4.1 Step 1: CAS sends check registration status request after web identification request

The Relying party’s signing application (CAS) sends check registration status request using /v2/identification/web/sessions/by-otc-request/{registrationId}/status function (getRegistrationStatus). CAS specifies registrationId which is in the response of /v2/identification/web/sessions/by-otc-request/{websessionId} function (createRegistration). As a result of this function CAS receives response with the current status of the registration.

URL:https://cqes-rpuat.b-trust.bg//signing-api/v2/identification/web/sessions/by-otc-request/{registrationId}/status
METHOD: GET
REQUEST HEADERS

KEY	VALUE	MANDATORY FIELD
Accept-language	bg || en	false
relyingPartyID	123456789	true
accept	application/json	true
Content-Type	application/json	true

REQUEST PARAMETERS

PATH:

- registrationId // Registration identifier – part of result of the synchronous operation /v2/identification/web/sessions/by-otc-request/{websessionId}

RESPONSE HEADERS

KEY	VALUE
Content-Type	application/json

RESPONSE BODY

Status code 200

```
{
   "createdOn": "2022-02-16T16:36:43.616Z", // registration request creation date
   "processedOn": "2022-02-16T16:36:43.616Z", // registration request processed date
   "registrationId": "string", // registration identifier
   "reviewStatus": "string", // registration status
   "reviewedOn": "2022-02-16T16:36:43.616Z", //registration reviewed date
   "status": "string", // registration status
   "userMessageBg": "string" // optional user message about the registration
}
```

Possible **Status** values:

- STARTED – the registration is started

- COMPLETED – all data needed for registration is gathered
- FINALIZED – the registration is confirmed – the client is registered
- REJECTED - the registration is rejected
- EXPIRED - the registration is expired

Enum:[STARTED, COMPLETED, FINALIZED, REJECTED, EXPIRED]

Possible **reviewStatus** values (This is the agent's review status):

- N_A – the registration is not completed
- FINALIZE_REJECT - the registration should be approved or redjected by agent
- ALLOW_DENY - the registration should be approved or redjected by agent after its automatic approval
- ALLOWED - the registration is approved by agent
- DENIED - the registration is rejected by agent

Enum:[N_A, FINALIZE_REJECT, ALLOW_DENY, ALLOWED, DENIED]

While the registration is in status STARTED its reviewStatus is N_A. If the registration should be approved by agent (not automated) – its reviewStatus is FINALIZE_REJECT. If the registration is approved automatically then reviewStatus is ALLOW_DENY. After the registration is reviewed by agent it could become with review status ALLOWED or DENIED

Status code 400, 401, 404, 500

5 Basic Scenario 4 – Check the status of web identification by result id after WEB identification request

5.1 Step 1: CAS sends check identification status request after web identification request

The Relying party's signing application (CAS) sends check registration status request using /v2/identification/web/{resultId}/status function (getWebIdentificationStatus). CAS specifies resultId which is in the response of /v2/identification/web/websession/start function (startWebIdentification). As a result of this function CAS receives response with the current status of the web identification.

URL: /v2/identification/web/{resultId}/status
METHOD: GET
REQUEST HEADERS

KEY	VALUE	MANDATORY FIELD

Accept-language	bg || en	false
relyingPartyID	123456789	true
accept	application/json	true
Content-Type	application/json	true

REQUEST PARAMETERS

PATH:

- resultId// Result identifier – part of result of the synchronous operation /v2/identification/web/websession/start

RESPONSE HEADERS

KEY	VALUE
Content-Type	application/json

RESPONSE BODY

Status code 200

```
{
   "data": {
      "status": "START_WEB_SESSION",
      "step": "WEB_IDV_REGISTRATION"
   },
   "responseCode": "OK",
   "code": "OK",
   "message": "Заявката е успешно изпълнена."
}
```

, where:

- **status** - current status of the request. Depending on the use-case this field can have values:
  - ON_BOARDING – started client onboarding
  - REGIX – client is being validate across national registries
  - ISSUED_CERTIFICATE – client certificate has been issued

  - IN_PROGRESS – identification request is in progress
  - SIGNING_CLIENT_DOCUMENT – client’s documents are in process of signing
  - SIGNED_CLIENT_DOCUMENT - client’s documents has been signed
  - AUTOMATIC_AGENT_SIGNING_CLIENT_DOCUMENT – client’s identification certificate is in process of sealing with BORICA(identification provider) seal
  - AUTOMATIC_AGENT_COMPOSE_CLIENT_DOCUMENT – identification certificate is in process of composing
  - AUTOMATIC_AGENT_SIGNED_CLIENT_DOCUMENT- client’s identification certificate has been sealed with BORICA(identification provider) seal
  - PROCESS_IS_FINISHED – identification process is finished
  - START_WEB_SESSION – web session is started
  - CREATE_SIGN_SESSION_REQUEST – sign session is in creation process
  - SIGN_SESSION – sign session is created
  - PROCESS_IS_CANCELED_BY_USER – identification process is canceled by user
  - IDENTIFICATION_REQUEST_DATA_AS_JSON – identification data as JSON is created

- **step** – current executing step of the current status. One status can have multiple steps to be executed. Depending on the use-case this field can have values:
  - PKCS10_REQ_GENERATED – certificate signing request(CSR) is generated
  - SIGNING_CLIENT_DOCUMENTS – in process of signing client’s documents
  - BORICA_SEAL_FOR_CLIENT_DOCUMENTS – in process of sealing identification certificate
  - UPDATE_DOCUMENTS_WITH_CERTIFICATE_DATA_FAILED – add client’s data in identification certificate
  - WEB_IDV_REGISTRATION – video identification is started
  - WEB_IDV_REGISTRATION_FAILED – video identification failed
  - START – process started
  - LOGIN – validating OTP code (received by client via SMS)
  - OTP – generating/sending OTP
  - CREATE_SIGN_SESSION_ID – creating sign session ID

- SIGN_SESSION_ID_SUCCESSFULLY_CREATED - sign session ID has been successfully generated
- SIGNING_WITH_SIGN_SESSION_SUCCESS – client's documents are successfully signed

Status code 400, 401, 404, 500