



API Documentation

Table of Contents

Documentation Changelog	5
1 Overview	6
1.1 Target Audience	6
1.2 Definitions	6
1.3 Preconditions.....	6
1.4 Subdomains	6
2 Lifecycle of an Identification.....	7
2.1 Overview.....	7
2.2 Preliminary vs. Final Results	8
3 Creating Identifications Using REST API.....	9
3.1 Overview.....	9
3.2 Details.....	9
3.2.1 Protocol	9
3.2.2 Host	9
3.2.3 Path.....	9
3.2.4 Version.....	10
3.2.5 Company ID	10
3.2.6 Transaction Number	10
3.2.7 Header	10
3.2.8 Body	10
3.2.9 Pre-defining Values for Questions.....	12
3.2.10 Result	13
3.2.11 Example	13
4 Creating Identifications Using Userdata Webform.....	14
4.1 Example of the Webform Userdata.....	14
4.2 Activate the Webform Userdata	14
4.3 URL to the Form	14
4.4 Explanation of the Parameters.....	14
5 Creating Identifications Using GET Requests	16
5.1 Example Link.....	16
5.2 Encoding	16
5.3 Maximum Length.....	17
5.4 Building the Link	17
5.4.1 Step 1: Generate the Path for the Request	17

5.4.2	Step 2: Generate the Un-encoded Query.....	17
5.4.3	Step 3: Sign the Un-encoded Query	19
5.4.4	Step 4: URL-encode the Query	20
5.4.5	Step 5: Build the Final URL	20
5.4.6	Step 6: Show Link to the User	21
6	Performing an Identification.....	22
6.1	Redirecting the User to the Identification Process	22
6.1.1	Linking to IDnow.....	22
6.1.2	Embedding the Identification Process.....	22
6.1.3	Campaign Tracking	23
6.2	Client Redirect URLs after Identification	23
6.2.1	Summary.....	23
6.2.2	Data	24
6.2.3	Example for Typical Usage.....	24
6.3	Mobile SDK	24
6.4	Estimated Waiting Time	24
6.4.1	Calculation of the Waiting Time	25
7	Identifications with eSigning	26
7.1	Document Definitions.....	26
7.1.1	Fields of a Document Definition	26
7.1.2	Document Signatures	27
7.1.3	Create New / Update Existing Document Definition.....	29
7.1.4	List Document Definitions	30
7.1.5	Get Single Document Definition	30
7.1.6	Upload Default Document.....	31
7.1.7	Download Default Document.....	31
7.2	Signing Documents.....	32
7.2.1	Listing Documents	32
7.2.2	Get a Single Document	33
7.2.3	Updating a Single Document	34
7.2.4	Upload Document to be Signed Using REST	34
7.2.5	Upload Document to be Signed Through File Upload Page	35
7.2.6	Download Signed Document Using REST	36
7.2.7	Download Signed Document Using SFTP / Encrypted Email / Encrypted ZIP	36
8	Retrieving Data via the REST API.....	37
8.1	Overview.....	37
8.1.1	Definitions	37

8.1.2	Preconditions.....	37
8.2	How to Access the Server	37
8.2.1	Hosts	37
8.3	Example Requests.....	38
8.3.1	Logging in.....	38
8.3.2	Retrieving a List of Identifications	39
8.3.3	Retrieving a Single Identification.....	40
8.3.4	Deleting an Identification	40
8.3.5	Copying an Identification.....	41
9	Retrieving Data via the SFTP	42
9.1	How to Access the Server	42
9.1.1	Clients	42
9.1.2	Production / Test Environment	42
9.1.3	Credentials.....	42
9.2	Example Session	43
9.2.1	Logging in.....	43
9.2.2	Listing Identifications.....	43
9.2.3	Downloading an Identification	44
9.2.4	Deleting an Identification	44
10	Retrieving Data via Mail	45
10.1	Mail with Encrypted ZIP (AES 256)	45
10.2	Encrypted Mail with ZIP (S/MIME)	45
10.3	Mail with Download Link.....	45
11	Retrieving Data via the Secure Download Webform	46
12	Errors	47
13	Result Data	49
13.1	Data Fields	49
13.1.1	Section Identification Process	49
13.1.2	Section Custom Data	50
13.1.3	Section Contact Data	51
13.1.4	Section User Data	51
13.1.5	Section Identification Document.....	53
13.1.6	Section Attachments	54
13.1.7	Section Questions.....	55
13.2	Result Attachments	55
13.2.1	Image Format	55

13.2.2	Audio Log.....	55
13.3	XML Result.....	56
13.3.1	XML Result Signature.....	57
13.4	JSON Format.....	57
14	Webhooks	60
14.1	Summary.....	60
14.2	Timing.....	60
14.3	IPs	60
14.4	Realtime / Final Webhooks	61
14.4.1	JSON Content.....	61
14.4.2	Example for Typical Usage.....	61
14.5	Aborted / Canceled Webhooks	61
14.5.1	JSON Content.....	61
14.5.2	Example for Typical Usage.....	62
14.5.3	Reasons for Failure	62
15	Testing.....	64
15.1	Selecting a Test Scenario	64
15.2	Automated Tests	65
15.2.1	Path.....	65
15.2.2	Header	65
15.2.3	Body.....	65
15.2.4	Example	65
15.3	Manual Test.....	66
15.4	Test with an IDnow Agent	66
15.5	Checking the Results.....	67

Documentation Changelog

VERSION 2016-02-15

Chapter	Description
1.4	Added overview of subdomains
3.2.6	Clarified description of transaction number (allowed characters, mentioning token)
7.1.2	Added several descriptions and signature placement/acrofield chapters in eSigning
7.2.3	Added information for updating documents
8.3.2	Added URL parameters for retrieving identifications via REST (pending, cancelled, aborted)
9.2.1	Added information about setup and use of private/public key pair authentication with SFTP
12	Added missing error codes and their description
14.3	Added IP addresses of webhooks

1 Overview

With the API companies can access the results of their identification processes.

The API can be used in various ways to access the identification results:

- Manually through secure download form
- Manually through a SFTP client
- Programmatically through a SFTP API
- Programmatically through a REST API

1.1 Target Audience

The document is targeted to developers of the third-party/company wanting to integrate the IDnow verification service into their own applications.

1.2 Definitions

Term	Definition
Company	Customer of IDnow who requests the identification (a bank for example).
User	The person who needs to be identified (end consumer).
Transaction Number	ID used to identify the requested identification. This ID should be used by the company as a key to assign the identification to an internal customer data set.

1.3 Preconditions

During setup, you should have received two values from IDnow:

Term	Definition
companyid	Uniquely identifies your company.
apiKey	This key is used to sign the request. Keep it secret and never use this on the client side.

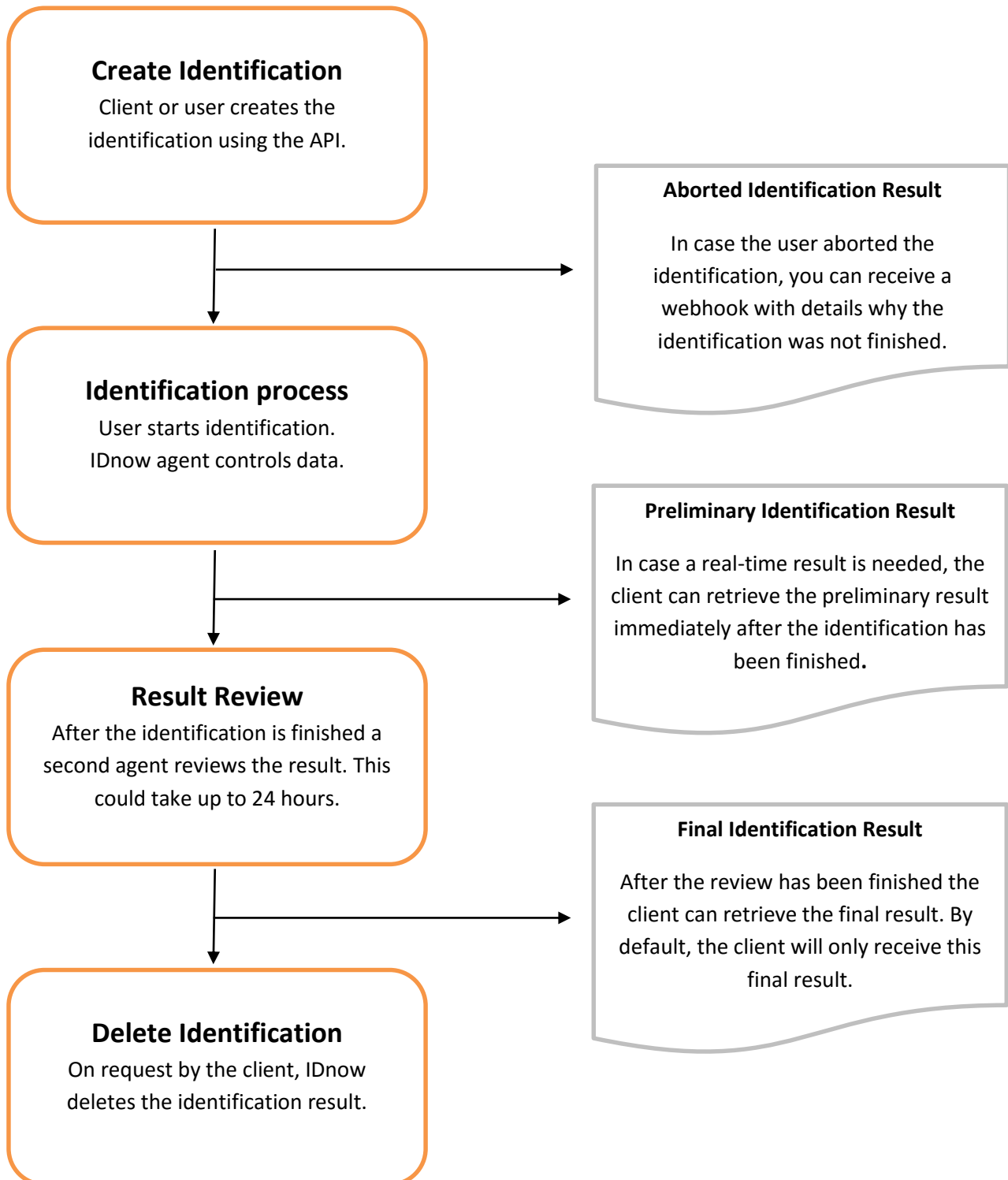
1.4 Subdomains

IDnow uses several subdomains for different services:

Term	Definition
go.idnow.de	Interface/Website the user sees, when identifying for a product of a company
gateway.idnow.de	Interface for companies to send client data to IDnow (before identification) or to retrieve data from IDnow (after identification)
api.idnow.de	Interface for companies to test their implementation

2 Lifecycle of an Identification

2.1 Overview



2.2 Preliminary vs. Final Results

If the company journey depends on the identification result and should continue in real-time it is recommended to use the preliminary results. In other cases, it is recommended to use the final results only.

Company Journey depends on identification result	Company Journey continues in real-time	Recommendation	Example
Yes	No	Final results only	User opens a bank account. After the identification, the user is asked to send in a paper contract to the bank. The account is opened after the paper contract is received by the bank.
No	Yes	Final results only	User opens a bank account. After the identification the user can directly log into his account and use basic features. The identification result is double checked by the bank in a manual (non real-time) process to activate/deactivate all features of the account.
Yes	Yes	Preliminary results and final results	User applies for a bank account. After the identification the bank account can directly be used in real-time. In case there are data changes detected during review, the bank account data can be updated automatically.

Note that preliminarily successful identifications can later be canceled in case they do not pass the review. Due to legal requirements, all personal information (incl. images, audio file, etc.) is deleted in such a case. If you decide to keep the customer nonetheless, it is at your own risk. However, you need to make sure that you retrieve the data from the preliminary results before it is deleted. Ideally, after you received the relevant webhook.

3 Creating Identifications Using REST API

3.1 Overview

The REST API lets you pass data about the user to IDnow's gateway server in order to then start an identification process.

The general flow is as follows:

- Your application collects the personal information in the normal checkout process.
- Your server creates a unique transaction number (here referred to as <transactionnumber>)
- Your server POSTs the data to IDnow's gateway server
- Your server redirects the client to IDnow's web server or you start the identification using the SDK (iOS / Android).
- The user is taken to the IDnow System and follows the verification steps.

To create an identification with eSigning, the respective documents need to be uploaded. For details on eSigning look up chapter Identifications with eSigning.

3.2 Details

3.2.1 Protocol

Live / Test server	https
--------------------	-------

To ensure that all parameters are encrypted, all request to the IDnow live server are always performed using HTTPS.

3.2.2 Host

Live server	gateway.idnow.de
Test server	gateway.test.idnow.de

To ensure that all parameters are encrypted, all requests to the IDnow server are always performed using HTTPS. HTTP is not allowed. If white labeling is used, the host depends on the setup of the company. You will be provided the URL during account setup.

3.2.3 Path

After creating a unique transaction token for the identification POST to:

```
/api/<version>/<companyid>/identifications/<transactionnumber>/start
```

Example: /api/v1/company-xyz/identifications/1234567890/start

3.2.4 Version

v1

Used to allow future evolutions of the API. Use "v1" for now.

3.2.5 Company ID

As provided by IDnow during account setup

The company ID uniquely identifying your company as provided by IDnow during your account setup.

3.2.6 Transaction Number

A unique ID generated by your system for each identification. This number will be provided back to you as your reference number, so that you can easily make a connection between your application and the identification. Typical examples are your internal contract numbers, user IDs or application numbers.

Though, to be compatible with the IDnow system, your transaction number may contain only the following characters:

a-zA-Z0-9_-

Additionally, the IDnow system (internally) assigns each identification another ID which is called token and also visible for the user. It takes the form of "ABC-DEFGH".

3.2.7 Header

Field	Mandatory	Content	Description
X-API-KEY	Yes	Your API key	Causes a 401 unauthorized error if not provided.
Content-Type	Yes	application/json	

3.2.8 Body

Use the POST request's body to send the user's details. The data has to be sent as UTF-8 encoded JSON. The expected JSON structure is flat with the following possible attributes (all of them being optional):

Parameter	Mandatory	Description	Example
birthday	No	The user's birthday in ISO 8601 format: YYYY-MM-DD	1975-12-20
birthplace	No	The user's birthplace	München

birthname	No	The user's birthname. Do not include prefixes like "Geb." Or "Geborene".	Meier
city	No	The user's city	München
country	No	The user's country. Uppercase two-letter code as defined in ISO 3166	DE
custom1	No	Custom text field. Use this to pass your own IDs, tags etc. You will get this information back in the identification results	Your own internal ID (e.g. 287492_23552)
custom2	No	See explanation for field custom1	
custom3	No	See explanation for field custom1	
custom4	No	See explanation for field custom1	
custom5	No	See explanation for field custom1	
trackingid	No	Custom tracking field. Can be used to pass tracking information. You will get this information back in the identification results. Can also be set as parameter „tid“ in user frontend.	
email	No	The user's email address	sampleuser@example.com
firstname	No	The user's first name(s)	Michael
gender	No	The user's gender. use 'MALE' or 'FEMALE'	MALE
lastname	No	The user's lastname	Berger
mobilephone	No	The user's mobile phone number. If no country code is given, 0049 is assumed.	0151 23411232
nationality	No	The user's nationality. Uppercase two-letter code as defined in ISO 3166	DE
street	No	The user's street	Bahnstrasse
streetnumber	No	The user's street number. This field can be configured to be part of the field "street", if you have street and number saved in one field in your database. If you wish to activate this setting please contact your technical account manager at IDnow.	27
title	No	Academic title. This should only be used, if the title is part of the name and shown in ID documents.	Dr.
zipcode	No	The user's zip code	80127
questions	No	Pre-defined values for questions shown to the identification agent.	See object "questions" below.

3.2.9 Pre-defining Values for Questions

IDnow supports asking additional questions during the identification process. The question is only shown to the identification agent. Questions can be in the form of radio buttons, dropdowns, input fields, date fields etc. Additionally, questions can be configured to be read-only or only be shown depending on the selection from other questions or depending on values from the identification itself (like country of the user for example).

Using this REST API, values for questions can be pre-defined for the agent. The agent will see the selected answer and will be able to modify it (unless read-only is enabled).

To pre-define the answers, set an array with the questions key and the desired value:

```
"questions": {
  "question_key_str": {
    "value": "value"
  },
  "question_key_int": {
    "value": 1
  },
  "question_key_date": {
    "value": "1975-12-20"
  }
}
```

The type of the value depends on the type of the question:

Question Type	Description	Value
RADIO_BOOLEAN	Input which allows to answers ("Yes / No", "True / False")	Stored as integer (0 = false, 1 = true)
RADIO_STRING	Multi-selection shown as radio buttons	String
DROPDOWN	Multi-selection shown as dropdown	String
DROPDOWN_COUNTRIES	Country selection dropdown	String, uppercase two-letter code as defined in ISO 3166
INPUT	Text input field	String
NUMERIC	Numeric input field	Integer
DATE	Date picker	The user's birthday in ISO 8601 format: YYYY-MM-DD

3.2.10 Result

In the response, you will get the following HTTP status codes for success:

Http Code	Message	Possible cause
201	Created	The identification was successfully created.
200	Ok	The identification already existed with this transaction number and has been updated with the new values provided. As long as the identification is not finished, you can update the data of the identification. If you try to update the identification after it has been completed, you will get an error with status code 409 (see below).

In addition, you will also get a POST body containing IDnow's unique internal id for this identification:

```
HTTP status code: 201

{
  "id": "IBA-H5FD8"
}
```

3.2.11 Example

The following examples show how you can create a new identification using curl. This example assumes that you received the following credentials during setup:

companyid	ihrebank
apiKey	exampleApiKey

The example request:

```
curl -i --header "X-API-KEY: exampleApiKey " --header "Content-Type: application/json" -d '{"birthday": "1975-12-20", "birthplace": "München", "city": "München", "country": "DE", "custom1": "287492_23552", "email": "sampleuser@example.com", "firstname": "Michael", "lastname": "Berger", "mobilephone": "0151 23411232", "nationality": "DE", "street": "Bahnstrasse", "streetnumber": "27", "zipcode": "80127"}' https://gateway.test.idnow.de/api/v1/ihrebank/identifications/1234567890/start
```

If called with valid credentials, the server will respond:

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Content-Length: 18

{"id": "ATK-EFXAL"}
```

4 Creating Identifications Using Userdata Webform

If you do not have the possibility to pass data to IDnow, then you can use a static link to a webform that is hosted by IDnow. In this webform the user will be prompted to enter his personal data that is used during the identification.

To create an identification with eSigning, the respective documents need to be uploaded. This can also be done by the user via in the webform. For details on eSigning look up chapter Identifications with eSigning.

4.1 Example of the Webform Userdata

An example of the landing page can be found at <https://go.idnow.de/video-demo/userdata>

4.2 Activate the Webform Userdata

To activate this feature of the API please inform your technical account manager at IDnow.

4.3 URL to the Form

You can either call the webform with a static link (no parameters) or by passing a reference number as a parameter. You can use the reference number after the identification to easily match the identification result with your internal processes (e.g. contract application).

Option 1: Static Link

Use this link if you cannot attach parameters to links or if you do not have a reference number of the user available.

```
<protocol>://<host>/<companyid>/userdata
```

Example: <https://go.idnow.de/video-demo/userdata>

Option 2: Link with Transaction Number Parameter

Use this link if you can attach parameters and you have a reference number of the user or contract application.

```
<protocol>://<host>/<companyid>/userdata/<transactionnumber>
```

Example: <https://go.idnow.de/video-demo/userdata/123456>

4.4 Explanation of the Parameters

Protocol

Live / Test server	https
--------------------	-------

To ensure that all parameters are encrypted, all request to the IDnow live server are always performed using HTTPS.

Host

Live server	go.idnow.de
Test server	go.test.idnow.de

If white labeling is used, the host depends on the setup of the company. You will be provided the URL during account setup.

Company ID

As provided by IDnow during account setup

The company ID uniquely identifying your company as provided by IDnow during your account setup.

Transaction Number

This number will be provided back to you as your reference number, so that you can easily make a connection between your application and the identification. Typical examples are your internal contract numbers, user IDs or application numbers.
--

See also 3.2.6 for allowed characters.

5 Creating Identifications Using GET Requests

To pass the user's data to IDnow, you have to generate a special link, which has to be opened by the user via browser.

The most common flow is as follows:

- Your application collects the personal information in the normal checkout/application process.
- When you want to trigger the identification, your application creates the special link described in this document. This link is generated on the server-side and includes security measures to prevent malicious use.
- The link is presented to the user as a button or text link "Start identification".
- The link passes the data to IDnow and starts identification process.

The generated link uses query authentication and follows the recommendations made by George Reese in [Principles for Standardized REST Authentication](#).

Summary of the principles:

All queries must be authenticated by signing the query parameters sorted in lower-case, alphabetical order using the private credential as the signing token. Signing should occur before URL encoding the query string

Since this method transmits user data via the URL, we recommend using REST or the Userdata webform over GET. Also, if you want to implement an eSigning identification, look up chapter Identifications with eSigning for details.

5.1 Example Link

The following link is a working example. It shows the most complete webform of the URL containing all possible parameters.

<https://go.test.idnow.de/v1/video-demo/start?birthday=1980-01-01&birthplace=M%C3%BCnchen&city=M%C3%BCnchen&country=DE&custom1=custom-code-1&custom2=custom-code-2&custom3=custom-code-3&custom4=custom-code-4&custom5=custom-code-5&email=info%40idnow.de&firstname=Max&lastname=Mustermann&mobilephone=%2B491761234567&nationality=DE&street=Musterstra%C3%9Fe+123×tamp=1&transactionnumber=XYZ-123&zipcode=12345&signature=ca571f42ee80dbf7442def87833b41edf35da7247ccf897ed91c72310db3bc35>

5.2 Encoding

All content needs to be UTF-8 encoded. This is especially important because the encoding will influence the calculation of the security token.

5.3 Maximum Length

Please note that browsers have maximum a length of 1040 characters for GET requests. If your request will be longer, please use the REST-version of this API.

5.4 Building the Link

5.4.1 Step 1: Generate the Path for the Request

The path part of the verification link consists of static parts combined with your company ID:

<code><protocol>://<host></code>
Example: https://go.idnow.de

Protocol

Live / Test server	https
--------------------	-------

To ensure that all parameters are encrypted, all request to the IDnow live server are always performed using HTTPS.

Host

Live server	go.idnow.de
Test server	go.test.idnow.de

If white labeling is used, the host depends on the setup of the company. You will be provided the URL during account setup.

5.4.2 Step 2: Generate the Un-encoded Query

The query that has to be signed is in the following format:

<code>/<version>/<companyid>/start?<queryDataString></code>
Example: <code>/v1/company-xyz/start?birthday=1975-12-20&firstname=Michael Peter&lastname=Bürger&email=michael@buerger.de&timestamp=1405074349&transactionnumber=1234567890</code>

Version

v1

Used to allow future evolutions of the API. Use "v1" for now.

Company ID

As provided by IDnow during account setup

The company ID uniquely identifying your company as provided by IDnow during your account setup.

QueryString

Build a query string by combining all values you want to send (plus timestamp). Pay attention to the following rules:

- Use the parameter names exactly as shown in the table below
- Add the parameters in alphabetical order (of the key) to the path
- Do not URL-encode the parameter values yet

Parameter	Mandatory	Description	Example
birthday	No	The users birthday in ISO 8601 format: YYYY-MM-DD	1975-12-20
birthname	No	The user's birthname. Do not include prefixes like "Geb." Or "Geborene".	Meier
birthplace	No	The user's birthplace	München
city	No	The user's city	Berlin
country	No	The user's country. Uppercase two-letter code as defined in ISO 3166	DE
custom1	No	Custom text field. Use this to pass your own IDs, tags etc. You will get this information back in the identification results	Your own internal ID (e.g. 287492_23552)
custom2	No	See explanation for field <code>custom1</code>	
custom3	No	See explanation for field <code>custom1</code>	
custom4	No	See explanation for field <code>custom1</code>	
custom5	No	See explanation for field <code>custom1</code>	
trackingid	No	Custom tracking field. Can be used to pass tracking information. You will get this information back in the identification results. Can also be set as parameter „tid“ in user frontend.	
email	No	The user's email address	sampleuser@example.com
firstname	No	The user's first name(s)	Michael Peter
gender	No	The user's gender. use 'MALE' or 'FEMALE'	MALE
lastname	No	the user's lastname	Berger
mobilephone	No	The user's mobile phone number. If no country code is given, 0049 is assumed.	0151 23411232
nationality	No	The user's nationality. Uppercase two-letter code as defined in ISO 3166	DE

street	No	The user's street.	Bahnstrasse
streetnumber	No	The user's street number. This field can be configured to be part of the field "street", if you have street and number saved in one field in your database. If you wish to activate this setting please contact your technical account manager at IDnow.	27
timestamp	Yes	Your current server time; milliseconds passed since 00:00:00 Thursday, 1 January 1970. Used to prevent replay attacks.	1405074349567
title	No	Academic title. This should only be used, if the title is part of the name and shown in ID documents.	Dr.
transactionnumber	Yes	This number will be provided back to you as your reference number, so that you can easily make a connection between your application and the identification. See also 3.2.6 for allowed characters.	Reference number from your application (e.g. Antragsnummer, Vorgangsnummer, etc)
zipcode	No	The user's zip code	80127

5.4.3 Step 3: Sign the Un-encoded Query

In the next step you will compute a hash over the string computed in 5.4.2 using your `apiKey`. The hash has to be computed using the HMAC-SHA256 algorithm:

```
HMACSHA256(queryString from step1, privateKey) --> signature|digest
```

Pay attention that your HMAC-SHA256 computation leads to the same results as those mentioned in [Hash-based message authentication code](#).

If your backend processes are built in Java, you might use the following code snippet:

```
public String computeDigest(String msg, String keyString) {
    String digest = null;
    try {
        /* Initialize the SHA256 algorithm using the key. Here
        javax.crypto.* is used. */
        SecretKeySpec key = new SecretKeySpec((keyString).getBytes("UTF-8"), "HmacSHA256");
        Mac mac = Mac.getInstance("HmacSHA256");
        mac.init(key);

        /* Calculate the signature */
        byte[] bytes = mac.doFinal(msg.getBytes("UTF-8"));

        /* Convert to hexadecimal values. Here,
        org.apache.commons.codec.binary.Hex is used */
```

```

        digest = new String( Hex.encodeHex( bytes ) );
    } catch (Exception e) {
        e.printStackTrace();
    }
    return digest;
}

```

5.4.4 Step 4: URL-encode the Query

In this step you need to encode the parameter values of the query string you created in the previous step. Make sure that you only encode the values, not the URL-part, parameter names, "&" and "=".

Input	/v1/video-demo/start?birthday=1980-01-01&birthplace=München&city=München&country=DE&custom1=custom-code-1&custom2=custom-code-2&custom3=custom-code-3&custom4=custom-code-4&custom5=custom-code-5&email=info@idnow.de&firstname=Max&lastname=Mustermann&mobilephone=+491761234567&street=Musterstraße 123×tamp=1&transactionnumber=XYZ-123&zipcode=12345
Output	/v1/idnow/start? birthday=1980-01-01&birthplace=M%C3%BCnchen&city=M%C3%BCnchen&country=DE&custom1=custom-code-1&custom2=custom-code-2&custom3=custom-code-3&custom4=custom-code-4&custom5=custom-code-5&email=info%40idnow.de&firstname=Max&lastname=Mustermann&mobilephone=%2B491761234567&nationality=DE&street=Musterra%C3%9Fe+123×tamp=1&transactionnumber=XYZ-123&zipcode=12345

5.4.5 Step 5: Build the Final URL

In the final step you will concatenate the results of the previous steps to a complete url:

Link part	Definition
Path	Use the protocol, host as built in step 1 (e.g. https://go.idnow.de)
URL-encoded Query String	Use the URL-encoded query string as built in step 4 (e.g. /v1/video-demo/start?birthday=1980-01-01&birthplace=München&city=München&country=DE&custom1=custom-code-1&custom2=custom-code-2&custom3=custom-code-3&custom4=custom-code-4&custom5=custom-code-5&email=info@idnow.de&firstname=Max&lastname=Mustermann&mobilephone=+491761234567&street=Musterstraße 123×tamp=1&transactionnumber=XYZ-123&zipcode=12345)
Signature	Append '&signature=<signature-digest>' as built in step 3 to the query (e.g. &signature=ca571f42ee80dbf7442def87833b41edf35da7247ccf897ed91c72310db3bc35)

Resulting link example:

```
https://go.test.idnow.de/v1/video-demo/start?birthday=1980-01-01&birthplace=M%C3%BCnchen&city=M%C3%BCnchen&country=DE&custom1=custom-code-1&custom2=custom-code-2&custom3=custom-code-3&custom4=custom-code-4&custom5=custom-code-5&email=info%40idnow.de&firstname=Max&lastname=Mustermann&mobilephone=%2B491761234567&nationality=DE&street=Musterstra%C3%9Fe+123&timestamp=1&transactionnumber=XYZ-123&zipcode=12345&signature=ca571f42ee80dbf7442def87833b41edf35da7247ccf897ed91c72310db3bc35
```

5.4.6 Step 6: Show Link to the User

Display the link to the user in form of a link or button. A click on the button will pass the data to IDnow and start the identification process.

6 Performing an Identification

6.1 Redirecting the User to the Identification Process

If all went well you will receive http status code "201 Created" with a body according to 3.2.10 from IDnow's gateway server. After receiving this confirmation, you can redirect the user to IDnow's identification process:

The target host depends on the usage of white labeling. If no white labeling is used, the host will always be "go.idnow.de". If white labeling is used, the host depends on the setup of the company (e.g. the host could be "ident.ihrebank.de").

There are two ways to build the link. Both links are equivalent. The identification process can also be embedded as an iframe or opened as popup.

6.1.1 Linking to IDnow

6.1.1.1 Long Link Using <companyid> and <transactionnumber>

Live server	<code>https://go.idnow.de/<companyid>/identifications/<transactionnumber></code>
Test server	<code>https://go.test.idnow.de/<companyid>/identifications/<transactionnumber></code>

6.1.1.2 Short Link Using <id> as Returned in Result

Live server	<code>https://go.idnow.de/<id></code>
Test server	<code>https://go.test.idnow.de/<id></code>

6.1.2 Embedding the Identification Process

IDnow supports embedding the identification (go.idnow.de) as an iframe. For the following embedding options **only** the long link version is supported:

Live server	<code>https://go.idnow.de/<companyid>/identifications/<transactionnumber>/identification/start</code>
Test server	<code>https://go.test.idnow.de/<companyid>/identifications/<transactionnumber>/identification/start</code>

Here are some additional thoughts:

- Make sure the client's data was sent to IDnow before loading the iframe or opening the popup
- Be aware that the embedding only makes sense for clients using a desktop PC, because video streams on mobile browsers is not supported

6.1.2.1 In an Iframe

For iframes we recommend to use at least 600px in width and 800px in height for the page to be displayed properly:

```
<iframe src="url/To/Identification" width="600" height="800"></iframe>
```

6.1.2.2 In a Popup

It is also possible to open the identification in a popup. In Javascript use the following command:

```
window.open("url/To/Identification ", "_blank",  
"width=600,height=800,left=50,top=50,location=no,menubar=no,resizable=yes,scrollbars=yes,status=no,titlebar=no,toolbar=no");
```

6.1.3 Campaign Tracking

You can pass a tracking id in the browser. This can for example be used to track marketing campaigns. To use this feature, append the “tid” parameter to the user url like this:

Live server	<code>https://go.idnow.de/<companyid>/identifications/<transactionnumber>?tid=<your tracking id></code>
Test server	<code>https://go.test.idnow.de/<companyid>/identifications/<transactionnumber>?tid=<your tracking id></code>

Note: This also works for the short links or the userdata forms:

Short link	<code>https://go.idnow.de/<id>?tid=<your tracking id></code>
Userdata form	<code>https://go.idnow.de/<companyid>/userdata?tid=<your tracking id></code>

The content of the tracking id will be returned in the trackingid field in the results. If you also need the trackingid in the result PDF, please contact IDnow support.

6.2 Client Redirect URLs after Identification

6.2.1 Summary

When this feature is activated for your account, after each identification the user will be redirected to a custom URL on your web server.

To activate the feature please contact your technical account manager at IDnow.

6.2.2 Data

The URLs support a place holder for the field “transactionnumber”, so that you can match the redirect to your internal transaction number.

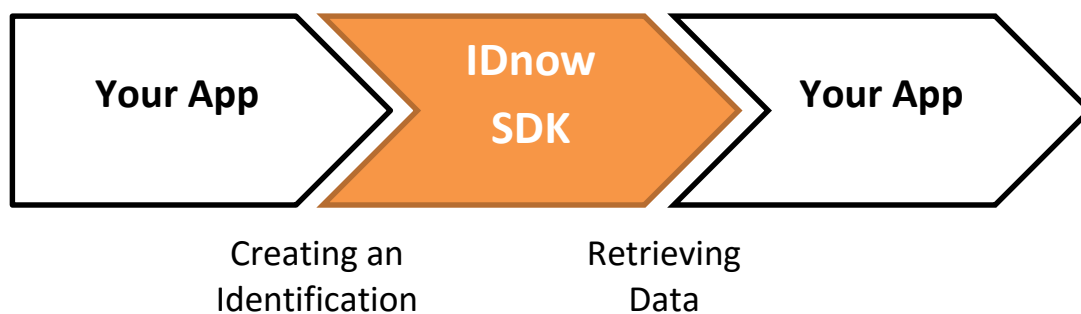
No additional data is passed to the redirect URL.

Example: After a successful identification the user is redirect to <https://www.yourcompany.com/ident-success?transaction-number=<transactionnumber>> where <transactionnumber> will be replaced by the actual transaction number of the identification.

6.2.3 Example for Typical Usage

After finishing the identification, the user is forwarded into a funnel on your website. The exact landing page depends on whether the identification has been successful or not.

6.3 Mobile SDK



The identification process can also be integrated in an Android or iPhone Application. For details see the respective SDKs:

<https://github.com/idnow/de.idnow.ios>

<https://github.com/idnow/de.idnow.android>

6.4 Estimated Waiting Time

You can access the current estimated waiting time for your account using the REST API. Make a GET request against /api/v1/<companyid> with the authToken obtained during login in the header. With the examples mentioned above, this will look like this:

```
curl -H "X-API-LOGIN-TOKEN: 7eb579f6-51b9-4a61-a32b-8bce0f84c6fa" -fO https://gateway.test.idnow.de/api/v1/ihrebank
```

The returned content will look like this:

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
```

Content-Length: 70

```
{
  "shortname": "ihrebank",
  "shortcode": "IBA",
  "name": "IhreBank AG",
  "estimatedWaitingTime": 30
}
```

The fields have the following meaning:

Parameter	Description	Example
shortname	The company ID of the account	Ihrebank
shortcode	A 3-digit shortcode of the account	IBA
name	The display name of the account	IhreBank AG
estimatedWaitingTime	The current estimated waiting time in seconds	30

6.4.1 Calculation of the Waiting Time

The estimated waiting time is calculated based on a sliding window of the last 10 minutes taking into account the current SLAs. Since the calculation is an estimate only rounded values 0, 15, 30, 45, 60, 120, 180 ... are returned. Additionally, a bit of buffer is added on top.

You can use the estimated waiting time to either display a note to users on your end or decide to use a waiting queue if load is exceptionally high (e.g. a marketing campaign has generated higher than forecasted demand).

NOTE: Since this calculation is an estimate, this should not be used to calculate SLA fulfillment. Instead, only the actual waiting times from the SLA report should be used.

7 Identifications with eSigning

Using the eSigning API, you can upload documents to be signed by the user during the identification. eSigning cannot be done without an identification.

Documents can be provided by:

- Uploading them using the REST API for each user (e.g. the contract of the user)
- Letting the user upload themselves using a file upload on the website of IDnow
- Providing a default document (e.g. T & C which do not change for each user)

7.1 Document Definitions

7.1.1 Fields of a Document Definition

Document definitions are configurations which tell the IDnow system, which documents are required from the user in order to carry out the eSigning identification. Also, for recurrent documents default files can be deposited.

Field	Mandatory	Content	Description
optional	No	false	Is the document required? If yes and the default document is provided, it will automatically be used
name	Yes	“Arbeitsvertrag”	The display name of the document
identifier	Yes	“document-abc”	The identifier of the document as used in the url. Only allowed a-z,-,_,
contentType	Yes	“application/pdf”	The mime type of the document
sortOrder	No	1	The order where to display the new document. The document will be inserted before this position. The order starts with 1.
version	No	V1.2.3	Optional field to store which version of the document this is.
signatures	No	{...}	See chapter 7.1.2 Fehler! Verweisquelle konnte nicht gefunden werden.

7.1.2 Document Signatures

To indicate that a contract was signed on the document itself, IDnow offers two possibilities to modify the document respectively:

- The user can "sign" the document by moving the mouse or his/her finger on the touchscreen (on mobile devices). This movement is captured and converted to a signature.
- The company can provide a "seal-like" image which will serve as a signature. Also, an IDnow default image can be used here.

The type of document signature can be defined under the "signatures" node within the document definition.

Field	Default	Description
position	{...}	Object to determine the position of the signature in the page or in the acrofield
type	ELECTRONIC	ELECTRONIC: seal with text HANDWRITTEN: written by user (feature has to be activated by IDnow)
signatureFontSize	8	Font size of the electronic signature (only necessary for type="ELECTRONIC")
signatureImage	idnow_seal	Background image for the electronic signature, a custom image can be provided by the company, allowed characters are: a-z0-9_- (only necessary for type="ELECTRONIC")
signatureImageScale	0.25	Scale of the background image (only necessary for type="ELECTRONIC")
optional	False	Is the signature required? If it is not optional and the page or acrofield is missing, there will be an error!

When using the default "idnow_seal" make sure, that there is enough space on the document to place it. It is 250px wide and 35px high.

For correct placement of the signature, the company needs to provide information where and how large the signature shall be on the document. This information is passed along in the "position" object. IDnow currently supports the following two ways of defining this placement.

7.1.2.1 Position Using Acrofields

An acrofield in the pdf file can be used. Here "left" and "bottom" are relative to borders of the acrofield.

Field	Description
acrofield	Name of the acrofield, in which the signature should be placed
left	Distance from the left border of the area of the specified acrofield
width	Width of the signature space
bottom	Distance from the bottom border of the area of the specified acrofield
height	Height of the signature space

7.1.2.2 Position Using Absolute Coordinates

A page of the pdf file can be specified. Here "left" and "bottom" are relative to borders of the page.

Field	Description
page	Number of page, on which the signature should be placed (indexing starts from 1)
left	Distance from the left border of the specified page
width	Width of the signature space
bottom	Distance from the bottom border of the specified page
height	Height of the signature space

7.1.2.3 Example for Acrofields with Seal as Signature

```
{
  ... document definition ...

  "signatures": {
    "mysignature1": {
      "position": {
        "acrofield": "my_acrofield_signature",
        "left": 50,
        "width": 200,
        "bottom": 50,
        "height": 35
      },
      "type": "ELECTRONIC",
      "signatureFontSize": 8,
      "signatureImage": "idnow_seal",
      "signatureImageScale": 0.25,
      "optional": "false"
    },
    "mysignature2": {
      ...
    }
  }
}
```

7.1.2.4 Example for Absolute Coordinates with Handwritten Signature

```
{
  ... document definition ...

  "signatures": {
    "mysignature1": {
      "position": {
        "page": "1",
        "left": 800,
        "width": 200,
        "bottom": 100,
        "height": 35
      },
      "type": "HANDWRITTEN",
      "optional": "false"
    },
    ...
  }
}
```

```
        "mysignature2": {  
            ...  
        }  
    }  
}
```

7.1.2.5 Overwriting Signatures for a Single User

You can define signatures at two levels:

- For all uploaded documents in the document definition object (see this chapter)
- For a single user for one document (see chapter 7.2.3)

If you use different names for the signatures, the signatures will be combined. If you use the same name for the signature, the signature for a single document will overwrite the signature of the document definition.

7.1.3 Create New / Update Existing Document Definition

7.1.3.1 Path

```
POST /api/<version>/<companyid>/documentdefinitions
```

```
Example: /api/v1/company-xyz/documentdefinitions
```

7.1.3.2 Example Request

```
curl -i --header "X-API-LOGIN-TOKEN: asdgg38434-34guzf3f4zf34fz34-ugfgu3u34"  
--header "Content-Type: application/json" -d "{\"optional\":false,  
\"name\":\"Arbeitsvertrag\", \"identifier\":\"doc1\",  
\"mimeType\":\"application/pdf\", \"sortOrder\":1}"  
http://gateway.test.idnow.de/api/v1/ihrebank/documentdefinitions
```

If called with valid credentials for an existing document definition:

```
HTTP/1.1 200 OK  
Access-Control-Allow-Origin: *  
Content-Type: application/json; charset=utf-8  
Content-Length: 2  
  
{}
```

If called with valid credentials for a new document definition:

```
HTTP/1.1 201 CREATED  
Access-Control-Allow-Origin: *  
Content-Type: application/json; charset=utf-8  
Content-Length: 2  
  
{}
```

7.1.4 List Document Definitions

7.1.4.1 Path

```
GET /api/<version>/<companyid>/documentdefinitions
```

```
Example: /api/v1/company-xyz/documentdefinitions
```

7.1.4.2 Example Request

```
curl -i --header "X-API-LOGIN-TOKEN: asdgg38434-34guzf3f4zf34fz34-ugfgu3u34"  
http://gateway.test.idnow.de/api/v1/ihrebank/documentdefinitions
```

If called with valid credentials, the server will respond:

```
HTTP/1.1 200 OK  
Access-Control-Allow-Origin: *  
Content-Type: application/json; charset=utf-8  
Content-Length: 2  
  
[{"optional":false,"name":"Arbeitsvertrag","identifier":"doc1","mimeType":"application/pdf","sortOrder":1}, {"optional":false,"name":"AGB","identifier":"doc2","mimeType":"text/plain","sortOrder":2}, {"optional":false,"name":"NDA","identifier":"doc3","mimeType":"application/xword","sortOrder":3}]
```

7.1.5 Get Single Document Definition

7.1.5.1 Path

```
GET  
/api/<version>/<companyid>/documentdefinitions/<documentDefinitionIdentifier>
```

```
Example: /api/v1/company-xyz/documentdefinitions/doc1
```

7.1.5.2 Example Request

```
curl -i --header "X-API-LOGIN-TOKEN: asdgg38434-34guzf3f4zf34fz34-ugfgu3u34"  
http://gateway.test.idnow.de/api/v1/ihrebank/documentdefinitions/doc1
```

If called with valid credentials, the server will respond:

```
HTTP/1.1 200 OK  
Access-Control-Allow-Origin: *  
Content-Type: application/json; charset=utf-8  
Content-Length: 2  
  
{"optional":false,"name":"Arbeitsvertrag","identifier":"doc1","mimeType":"application/pdf","sortOrder":1}
```

7.1.6 Upload Default Document

You can upload the document for a document definition which will be used as the basis when a new signature process is created. You can use this for documents which are the same for all users (e.g. Terms and Conditions which are not customized per user).

7.1.6.1 Path

```
POST
/api/<version>/<companyid>/documentdefinitions/<documentDefinitionIdentifier>/data
```

Example: /api/v1/company-xyz/documentdefinitions/doc1/data

7.1.6.2 Example Request

```
curl -i -X POST --data-binary @localfile.pdf --header "X-API-LOGIN-TOKEN: asdgg38434-34guzf3f4zf34fz34-ugfgu3u34" --header "Content-Type:application/octet-stream"
http://gateway.test.idnow.de/api/v1/ihrebank/documentdefinitions/doc1/data
```

If called with valid credentials, the server will respond:

```
HTTP/1.1 100 Continue

HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Length: 0
```

7.1.7 Download Default Document

7.1.7.1 Path

```
GET
/api/<version>/<companyid>/documentdefinitions/<documentDefinitionIdentifier>/data
```

Example: /api/v1/company-xyz/documentdefinitions/doc1/data

7.1.7.2 Example Request

```
curl -o download.pdf --header "X-API-LOGIN-TOKEN: asdgg38434-34guzf3f4zf34fz34-ugfgu3u34"
http://gateway.test.idnow.de/api/v1/ihrebank/documentdefinitions/doc1/data
```

If called with valid credentials, the server will respond:

% Total Current		% Received		% Xferd		Average Speed		Time	Time	Time	
						Dload	Upload	Total	Spent	Left	
Speed											
100	11	0	11	0	0	390	0	--:--:--	--:--:--	--:--:--	
407											

7.2 Signing Documents

For each identification / signing process, you can list the documents to be signed as well as upload documents to be signed and finally get the signed result of the identification.

Field	Mandatory	Content	Description
name	No	“Arbeitsvertrag”	The display name of the document. Will use the name of the document definition by default but can be overwritten.
version	No	V1.2.3	Optional field to store which version of the document this is. Will use the version of the document definition by default but can be overwritten.
hash	Read-Only	094c33504a...	The SHA256 hash of the document
displayHash	Read-Only	094c-3350-4ab6	The fingerprint of the document to display to the user. This are the first 12 bytes of the hash.
status	Read-Only	NEED_UPLOAD, UPLOADED, SIGNED	The status of the document
documentDefinition	Read-Only	Link to document definition object	
signatures	No	{...}	(optional) additional signatures

7.2.1 Listing Documents

The IDnow system generates an instance of the document definition for each identification which will then be used to perform the identification.

7.2.1.1 Path

```
GET
/api/<version>/<companyid>/identifications/<transactionnumber>/documents
```

Example: /api/v1/company-xyz/identifications/1234567890/documents

7.2.1.2 Example Request

You can list the documents which are going to be signed in this process:

```
curl -i --header "X-API-LOGIN-TOKEN: asdgg38434-34guzf3f4zf34fz34-ugfgu3u34"
http://gateway.test.idnow.de/api/v1/ihrebank/identifications/XFA-
DASD4/documents
```

If called with valid credentials, the server will respond:

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Content-Length: 2

[{"name":"Arbeitsvertrag","version":null,"hash":"094c33504ab62a49957ed5f051f
c10fc0e1bd7feadfdb223654270835e693bb4","displayHash":"094c-3350-
4ab6","status":"UPLOADED","documentDefinition":{"optional":false,"name":"Arb
eitsvertrag","identfier":"doc1","mimeType":"application/pdf","sortOrder":1}
}]
```

7.2.2 Get a Single Document

You can list the documents which are going to be signed in this process.

7.2.2.1 Path

```
GET
/api/<version>/<companyid>/identifications/<transactio
nnumber>/documents/<documentDefinitionIdentifier>
```

Example: /api/v1/company-xyz/identifications/1234567890/documents/doc1

7.2.2.2 Example Request

```
curl -i --header "X-API-LOGIN-TOKEN: asdgg38434-34guzf3f4zf34fz34-ugfgu3u34"
http://gateway.test.idnow.de/api/v1/ihrebank/identifications/XFA-
DASD4/documents/doc1
```

If called with valid credentials, the server will respond:

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Content-Length: 2

[{"name":"Arbeitsvertrag","version":null,"hash":"094c33504ab62a49957ed5f051f
c10fc0e1bd7feadfdb223654270835e693bb4","displayHash":"094c-3350-
4ab6","status":"UPLOADED","documentDefinition":{"optional":false,"name":"Arb
eitsvertrag","identfier":"doc1","mimeType":"application/pdf","sortOrder":1}
}]
```

7.2.3 Updating a Single Document

The IDnow system uses each document definitions set up to create "instances" when an identification is created. It is possible to modify these instances in certain limits before the identification is started. The elements "name", "version" and "signatures" (see also table in 7.2) can be updated. The parameters "name" and "version" will overwrite the one given from the document definition. The last parameter "signatures" takes a special place:

- A signature of the document definition can be overwritten if the respective identifier is used
- A signature with a new identifier can extend the scope of the signatures

7.2.3.1 Path

You can overwrite the document definition by posting the a JSON with the respective elements ("name", "version" and "signatures"). For details on their format and allowed values, see the previous chapters.

```
POST
/api/<version>/<companyid>/identifications/<transactionnumber>/documents/<documentDefinitionIdentifier>
```

Example: /api/v1/company-xyz/identifications/1234567890/documents/doc1

7.2.3.2 Example Request

Here the name is updated and a signature added. Note that "signatures" only defines the "position" object, the remaining settings will be done by the defaults.

```
curl -i --header "X-API-LOGIN-TOKEN: asdgg38434-34guzf3f4zf34fz34-ugfgu3u34"
--header "Content-Type: application/json" -d "{\"name\": \"My Extended
Contract\", \"signatures\": {\"mysignature3\": {\"position\":
{\"acrofield\": \"my_acrofield_signature_3\", \"left\": 50, \"width\": 200,
\"bottom\": 50, \"height\": 35 }}}}\"
http://gateway.test.idnow.de/api/v1/ihrebank/identifications/1234567890/documents/doc1
```

7.2.4 Upload Document to be Signed Using REST

The defined documents (without default file) need to be uploaded, in order to perform the eSigning identification. These are the documents which differ for each user, e.g. the contract likely has the user's name on it and therefore needs to be uploaded for every user. Note that identifier of a document definition therefore refers to a class of documents rather than to a specific document (because each user has a different name and therefore a different document of the same class, though).

7.2.4.1 Path

```
POST
/api/<version>/<companyid>/identifications/<transactionnumber>/documents/<documentDefinitionIdentifier>/data
```

Example: /api/v1/company-xyz/identifications/1234567890/documents/doc1/data

7.2.4.2 Example Request

```
curl -i -X POST --data-binary @localfile.pdf --header "X-API-LOGIN-TOKEN:
asdgg38434-34guzf3f4zf34fz34-ugfgu3u34" --header "Content-
Type:application/octet-stream" http://gateway.test.idnow.de/api/v1/ihrebank/
XFA-DASD4/documents/doc1/data
```

If called with valid credentials, the server will respond:


```
HTTP/1.1 100 Continue

HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Length: 0
```

7.2.5 Upload Document to be Signed Through File Upload Page

If you want the user to upload the document to be signed by himself, you can use our file upload page. For this, the user has to download the contract on your side, and upload the document on our side.

Einfach, sicher und bequem:
Identifizierung per Videochat



1. Upload Document

2. Check Document

3. Identification

4. Sign Contract

5. Result

NUMMER FÜR SMS TAN
+49 176 103 054 32

VORGANGSNUMMER
270 303 123

1

Upload Document

Please choose your document to upload. This should be the contract which was provided to you at the end of the signup process.

Choose Document

NOTE: You can only do this for one document (the first document definition as determined by the order field). If you create several document definitions and do not provide default document or upload using the REST API, the user will not be able to start the signature / identification process.

7.2.6 Download Signed Document Using REST

7.2.6.1 Path

```
POST
/api/<version>/<companyid>/identifications/<transactionnumber>/documents/<documentDefinitionIdentifier>/signed
```

Example: /api/v1/company-xyz/identifications/1234567890/documents/doc1/signed

7.2.6.2 Example Request

To get the final signed result call the following URL:

```
curl -o signed.pdf --header "X-API-LOGIN-TOKEN: asdgg38434-34guzf3f4zf34fz34-ugfgu3u34" http://gateway.test.idnow.de/api/v1/ihrebank/XFA-DASD4/documents/doc1/signed
```

If called with valid credentials, the server will respond:

% Total		% Received		% Xferd		Average Speed		Time	Time	Time
Current						Dload	Upload	Total	Spent	Left
Speed										
100	11	0	11	0	0	390	0	--:--:--	--:--:--	--:--:--
407										

7.2.7 Download Signed Document Using SFTP / Encrypted Email / Encrypted ZIP

The signed documents will be included in the result ZIP files as well using the name `<documentDefinitionIdentifier>_signed.pdf` (e.g. doc1_signed.pdf).

8 Retrieving Data via the REST API

8.1 Overview

The recommended way to retrieve identification data is via the REST API. For details on the format of the retrievable data, see chapter Result Data. Also, to implement an event-based data retrieval procedure, see the chapter Webhooks.

8.1.1 Definitions

companyId	A short alphanumeric value which uniquely identifies your company
apiKey	A secret value used for your authentication. It always stays the same
authToken	A token which you obtain after a successful login. Stays the same during a single session. Every login creates a new token. Invalidated if unused for an hour

8.1.2 Preconditions

During setup, you should have received the company ID and the API key.

8.2 How to Access the Server

The following applies to all requests made by companies:

- encoding UTF-8 is used
- the server's response contains JSON-formatted data

In general, obtaining data via REST is a 2-step-process:

- You POST to the login route (providing your company ID and API key) to obtain a valid authToken
- You access your data with GET-Requests which contain the authToken from step 1

8.2.1 Hosts

As hosts you can either use '*gateway.idnow.de*' or '*gateway.test.idnow.de*' depending on whether you want to access production data or test data.

While the test system can be accessed via http, the production system can only be accessed via https:

Envir.	Protocol	Host	Port	Example
Test	https	gateway.test.idnow.de	443	https://gateway.test.idnow.de/api/v1/ihrebank
Live	https	gateway.idnow.de	443	https://gateway.idnow.de/api/v1/ihrebank

8.3 Example Requests

The following examples show how you can:

- login to the server
- list your identifications
- access a zip file with detailed information of a single identification

All examples assume that you want to access your test data and that during setup you received the following credentials:

companyid	ihrebank
apiKey	BXCexampleexampleexampleexampleRP

8.3.1 Logging in

Before accessing your identification data, you have to obtain a valid authToken. To do so, make a POST request to `/api/v1/<companyid>/login` with a JSON request body containing your API key. With the examples mentioned above, this will look like this:

```
curl -i -H "Content-Type: application/json" -d '{"apiKey":  
:"BXCexampleexampleexampleexampleRP"}'  
https://gateway.test.idnow.de/api/v1/ihrebank/login
```

If called with valid credentials, the server will respond:

```
HTTP/1.1 200 OK  
Access-Control-Allow-Origin: *  
Content-Type: application/json; charset=utf-8  
Content-Length: 52  
  
{  
  "authToken": "7eb579f6-51b9-4a61-a32b-8bce0f84c6fa"  
}
```

The JSON response consists of a JSON-formatted object containing your 'authToken' which you will have to send with the following requests. The authToken will be invalidated for security reasons if it is not used for an hour.

The authToken will be invalidated if it is not used for an hour. If you do not use it for an hour you will get a 401 "Unauthorized" answer and you have to call login again.

If you provide a wrong company ID or a wrong API key the server will respond with *"401 Unauthorized"*.

8.3.2 Retrieving a List of Identifications

To retrieve a list of your identifications, make a GET request against `/api/v1/<companyid>/identifications` which contains the `authToken` obtained during login in the header. You can retrieve successful, pending, cancelled and aborted identifications:

1) Successful Identifications:

```
curl -i -H "X-API-LOGIN-TOKEN: 7eb579f6-51b9-4a61-a32b-8bce0f84c6fa"
https://gateway.test.idnow.de/api/v1/ihrebank/identifications
```

2) Pending Identifications:

```
curl -i -H "X-API-LOGIN-TOKEN: 7eb579f6-51b9-4a61-a32b-8bce0f84c6fa"
https://gateway.test.idnow.de/api/v1/ihrebank/identifications?pending=true
```

3) Cancelled Identifications:

```
curl -i -H "X-API-LOGIN-TOKEN: 7eb579f6-51b9-4a61-a32b-8bce0f84c6fa"
https://gateway.test.idnow.de/api/v1/ihrebank/identifications?cancelled=true
```

4) Aborted Identifications:

```
curl -i -H "X-API-LOGIN-TOKEN: 7eb579f6-51b9-4a61-a32b-8bce0f84c6fa"
https://gateway.test.idnow.de/api/v1/ihrebank/identifications?aborted=true
```

If you passed a valid login token, the server will respond for example with:

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Content-Length: 270

{
  "identifications": [{
    "identificationprocess": {
      "result": "SUCCESS",
      "agentname": "HMUELLER",
      "identificationtime": "2014-06-02T05:03:54Z",
      "type": "WEB",
      "transactionnumber": "AH73JK3LM",
      "companyid": "ihrebank",
      "id": "IBA-H7GB6",
      "filename": "AH73JK3LM.zip",
      "href":
"/api/v1/ihrebank/identifications/AH73JK3LM.zip"
    },
    <Rest omitted>
  }]
}
```


For a detailed documentation of the fields you get from the identifications array, please see chapter 13.1.1.

8.3.3 Retrieving a Single Identification

8.3.3.1 Retrieving a Single Identification as ZIP

Every identification obtained by listing your identification in the step above contains a 'href' attribute. You can use this absolute path to download detailed information for the identification.

```
curl -H "X-API-LOGIN-TOKEN: 7eb579f6-51b9-4a61-a32b-8bce0f84c6fa" -fO  
https://gateway.test.idnow.de/api/v1/ihrebank/identifications/AH73JK3LM.zip
```

The contents of the zip-file depend on your configuration (which is managed by IDnow). It might contain a PDF File, text files, image files and audio files.

8.3.3.2 Retrieving a Single Identification as JSON

It is also possible to retrieve the JSON of a single identification object. Make a GET request against /api/v1/<companyid>/identifications/<transactionnumber> with the authToken obtained during login in the header. With the examples mentioned above, this will look like this:

```
curl -H "X-API-LOGIN-TOKEN: 7eb579f6-51b9-4a61-a32b-8bce0f84c6fa" -fO  
https://gateway.test.idnow.de/api/v1/ihrebank/identifications/AH73JK3LM
```

Note that in comparison to the ZIP download the example URL above does not end in “.zip”. The content of the single identification result is the same as the JSON format described in chapter 13.1.

```
{  
  "identificationprocess": {  
    ...rest of JSON...  
  },  
  "customdata": {  
    ...rest of JSON...  
  },  
  ...rest of JSON...  
}
```

8.3.4 Deleting an Identification

To delete an identification, make a DELETE request against /api/v1/<companyid>/identifications/<transactionnumber> with the authToken obtained during login in the header. With the examples mentioned above, this will look like this:

```
curl -i -H "X-API-LOGIN-TOKEN: 7eb579f6-51b9-4a61-a32b-8bce0f84c6fa" -X  
DELETE  
https://gateway.test.idnow.de/api/v1/ihrebank/identifications/AH73JK3LM
```

If you passed a valid login token, the server will respond for example with:

```
HTTP/1.1 200 OK
```

8.3.5 Copying an Identification

To copy an identification to a different account, make a POST request against `/api/v1/<companyid>/identifications/<transactionnumber>/copy` with the `authToken` obtained during login in the header. The body must contain a JSON node with 2 elements:

target_companyid	The company ID of the account to copy to (e.g. "ihrebank")
target_transactionnumber	A new transaction number in the target account. Optional parameter. If not sent, the same transaction number is used.

The result will be 201 CREATED and will contain the new id generated by IDnow.

NOTE: Your account needs permissions to copy data to a different account. If the permission is not enabled or you copy to an account where you do not have the permissions, you will get a 403 FORBIDDEN. Contact acm@idnow.de if you want to set this up.

With the examples mentioned above, this will look like this:

```
curl -i -H "X-API-LOGIN-TOKEN: 7eb579f6-51b9-4a61-a32b-8bce0f84c6fa" -H
"Content-Type: application/json" -d "{\"target_companyid\" : \"ihrebank\",
\"target_transactionnumber\" : \"123456\" }"
https://gateway.test.idnow.de/api/v1/ihrebank/identifications/AH73JK3LM/copy
```

If called with valid credentials, the server will respond:

```
HTTP/1.1 201 CREATED
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Content-Length: 52

{
    "id": "CGD-DSFGE"
}
```

9 Retrieving Data via the SFTP

IDnow offers a SFTP (SSH File Transfer Protocol) gateway which can be used by IDnow's clients to access the identifications. SFTP is an encrypted communication protocol which can be used to list, retrieve and delete the results of identifications. For details on the format of the retrievable data, see chapter Result Data. Also, to implement an event-based data retrieval procedure, see the chapter Webhooks.

9.1 How to Access the Server

9.1.1 Clients

You can use any SFTP client to access the SFTP-Server. Common examples are:

- WinSCP
- Filezilla
- sftp / scp: command line tools

Beside these tools, any SFTP compatible client or library can be used. The command line tool will be used to illustrate the examples in this document.

9.1.2 Production / Test Environment

IDnow offers a production and a test environment. The production environment will hold the identification results from live identifications. The test environment can be used to securely test an implementation. This environment is completely separated from the production environment.

The SFTP servers on the production environment can be accessed on "gateway.idnow.de" port 22. The SFTP servers for the test environment can be accessed on "gateway.test.idnow.de" port 22.

9.1.3 Credentials

On setup, IDnow provides a company ID (a short alphanumeric value which uniquely identifies your company) and an API key. Your company ID will be used as the login name and the API key will be used as password. Please get in contact with your IDnow account manager if you didn't receive your company ID or API key yet.

It is also possible to access sftp with a private/public key pair. If you would like to use this possibility, generate a key pair and contact acm@idnow.de, so we can setup the public key on the IDnow server.

9.2 Example Session

9.2.1 Logging in

If you received a company ID of "ihrebank" this would start your session:

```
sftp -P 22 ihrebank@gateway.idnow.de
```

SFTP will respond with:

```
Password authentication
Password:
```

After entering your API key and pressing enter you should see:

```
Connected to gateway.idnow.de.
sftp>
```

In case you set up a key pair for authentication, the first and second snippet boils down to:

```
sftp -oIdentityFile=/path/to/private/keyfile 22 ihrebank@gateway.idnow.de
```

9.2.2 Listing Identifications

There is a single directory you read files from. Your client will be in this directory per default. Use `ls` inside the directory to list all available identifications.

```
sftp> ls
XAFV3SPMBOASY9HA.zip
UH56FGT4DHN783JD.zip
OIHI67ASZUFG2562.zip
...
sftp> ls -altr XAFV3SPMBOASY9HA.zip
-rw-rw-rw-  1 ihrebank ihrebank  4711 Jul 28 13:41 XAFV3SPMBOASY9HA.zip
```

As the file permissions indicate you can read or delete the file. Please don't rely on the file size. Since the files are generated when you access them, we cannot know the exact size in advance. A single zip file will be listed per finished identification.

Per default the filename is built from the identification's token. However, this default filename can be changed per company and might include your custom fields as well.

To avoid a server overloading the **output of the ls command is limited to 50 items**. This does not mean there are only 50 items on the server. To view other items, download and delete visible items.

9.2.3 Downloading an Identification

Use 'get <filename>' inside the SFTP client to download a file to your local directory:

```
sftp> get XAFV3SPMBOASY9HA.zip
Fetching /XAFV3SPMBOASY9HA.zip to XAFV3SPMBOASY9HA.zip
/XAFV3SPMBOASY9HA.zip
```

After doing that, the zip file will reside in your local directory. Since the file is generated on demand there is no risk to receive only parts of the zip file.

9.2.4 Deleting an Identification

Use 'rm <filename>' inside the SFTP client to delete an identification. IDnow holds backups for 90 days, so IDnow will completely remove the identification after this period (when it gets removed from the backups as well). Please note that IDnow might not have the possibility to restore an identification if you accidentally delete an identification.

```
sftp> rm XAFV3SPMBOASY9HA.zip
Deleting XAFV3SPMBOASY9HA.zip
```

10 Retrieving Data via Mail

To set up one of the following methods, contact us via acm@idnow.de. For details on the format of the retrievable data, see chapter Result Data.

10.1 Mail with Encrypted ZIP (AES 256)

If this option is used, the company needs to choose a password and communicate it to IDnow. Then for every identification, the results will be encrypted using the given password in a ZIP-file and sent via mail to the company. The encryption used is AES 256.

Note: To open the file on Windows use a dedicated tool, e.g. 7-Zip.

10.2 Encrypted Mail with ZIP (S/MIME)

If this option is used, the company needs to generate a private/public key pair and communicate the public key to IDnow. The following certificate formats (and respective file extensions) are supported:

- DER binary encoded X.509 (.der)
- Base64 encoded X.509 (.cer, .pem)
- PKCS#7 (.p7b)

Then for every identification, the results will be sent as ZIP-file via a **secure** mail. The mail as well as its attachment get encrypted.

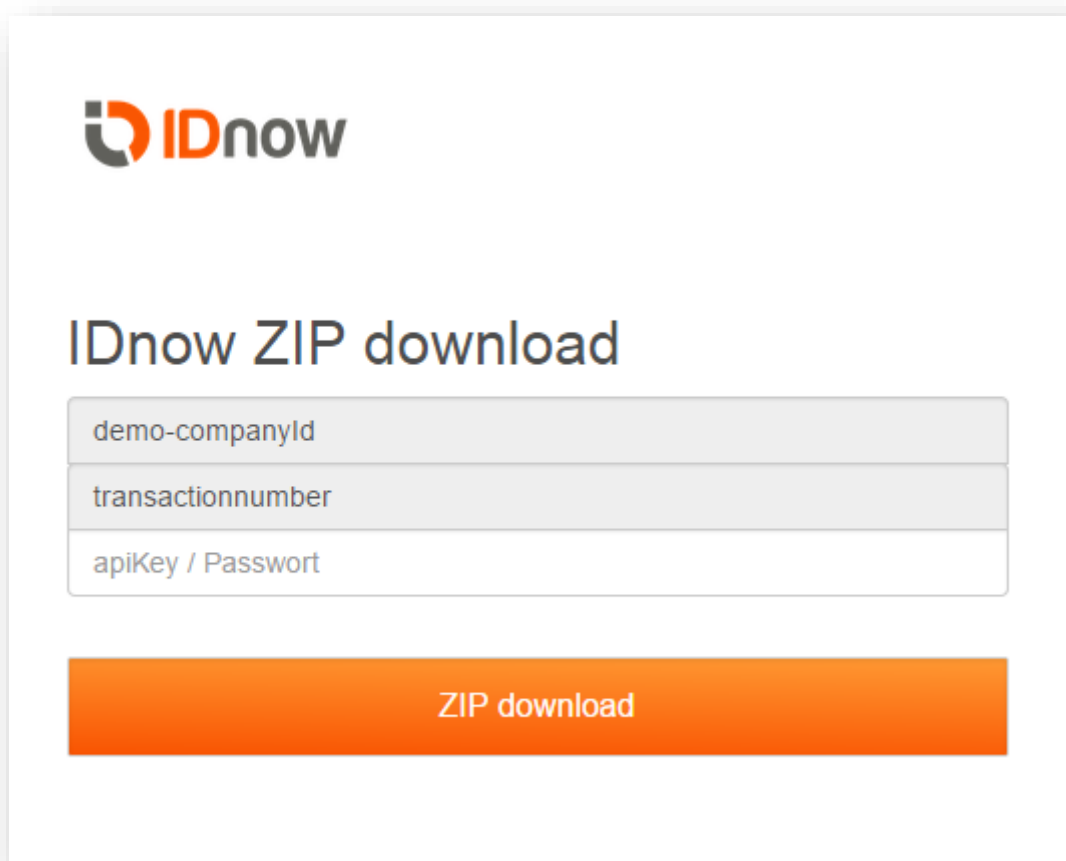
10.3 Mail with Download Link

With this option, you will receive a mail per identification which contains a link to the Secure Download Webform (Chapter 11), where the data can be downloaded using the respective credentials.

11 Retrieving Data via the Secure Download Webform

When this feature is activated for your account, after each identification you will receive an email with a link to the secure download form. In the download webform you will be prompted to enter your API key. After successful authentication a download of the identification result in ZIP format starts. For details on the format of the retrievable data, see chapter Result Data.

To activate the feature please contact your technical account manager at IDnow.



The image shows a web form titled "IDnow ZIP download". At the top left is the IDnow logo. Below the title, there are three input fields stacked vertically. The first field contains the text "demo-companyId". The second field contains the text "transactionnumber". The third field contains the text "apiKey / Passwort". Below these fields is a large orange button with the text "ZIP download" in white.

1: Example of secure download form

12 Errors

The following errors might occur when using the API:

HTTP Code	Message	Possible causes	Indicates wrong usage of the API
400	Bad Request	UNRECOGNIZED_FIELD UNPARSEABLE_JSON MISSING_TRANSACTION_TOKEN BAD_REQUEST	yes
401	Unauthorized	INVALID_LOGIN_TOKEN SECURITY_VIOLATION	yes
404	Not Found	OBJECT_NOT_FOUND	yes
409	Existing Identification	EXISTING_IDENTIFICATION	yes
412	Precondition Failed	PRECONDITION_FAILED	yes
500	Internal Server Error	INTERNAL_SERVER_ERROR	no

The following table contains details on the causes:

Cause	Description
UNRECOGNIZED_FIELD	Your request contains a field which is not recognized by the system.
UNPARSEABLE_JSON	The json body of your request is invalid. Therefore, it cannot be parsed.
MISSING_TRANSACTION_TOKEN	The transaction token is missing in your request. So the system does not know which identification object you try to access.
BAD_REQUEST	Your data was badly formatted in another way. E.g. wrong format for date or country field. Check the key and message of the response for details.
INVALID_LOGIN_TOKEN	You did provide a wrong API key in the request header.
SECURITY_VIOLATION	You tried to access an identification which does not belong to you.
OBJECT_NOT_FOUND	You used a wrong url. The identification or data you trying to get does not exist.
EXISTING_IDENTIFICATION	You tried to use an existing transaction number of an identification, which already existed and is finished. Updating finished identifications is not possible.
PRECONDITION_FAILED	The next step you request cannot be performed because something is missing. E.g. You try to run an eSigning identification, but forgot to upload the documents.
INTERNAL_SERVER_ERROR	Something unexpected did go wrong in the system. Not caused by wrong usage of the API, but by an error on the IDnow system.

If you get one of these errors, the server will send additional information in the response body. Those error responses all have the same structure.

An example for a missing or wrong API key:

```
HTTP status code: 401

{
  "errors": [{
    "cause": "INVALID_LOGIN_TOKEN",
    "id": "487800773",
    "key": null,
    "message": null
  }]
}
```

An example for a badly formatted date field:

```
HTTP status code: 400

{
  "errors": [{
    "cause": "BAD_REQUEST",
    "id": "73464627",
    "key": "birthday",
    "message": "Please provide a correct value for 'birthday'"
  }]
}
```

An example when trying to update a finished identification:

```
HTTP status code: 409

{
  "errors": [{
    "cause": "EXISTING_IDENTIFICATION",
    "id": "34253467",
    "key": null,
    "message": null
  }]
}
```

With the following meaning of the fields:

Attribute		Meaning
cause		short textual information about the error
id		The id of the exception. IDnow can use this id to check for the cause of an exception. If you need assistance with an exception, please contact IDnow and provide this id to the technical support.
key		might contain additional information about the field which cause the exception (for example: the field with a wrong format)
message		Human-readable error message in English

13 Result Data

Depending on the API used for retrieving the results, the result can either be returned as JSON (JavaScript Object Notation, see <http://en.wikipedia.org/wiki/JSON>) or XML.

13.1 Data Fields

The following tables shows the different sections and values of the result:

13.1.1 Section Identification Process

This section holds the result and general information about the identification process. This section is marked by the key “identificationprocess”.

Parameter	Mandatory	Description	Example
result	Yes	The result of the identification. Possible values are “SUCCESS”, “SUCCESS_DATA_CHANGED” and “FRAUD_SUSPICION_CONFIRMED”. For an explanation of the results states, please see below.	SUCCESS
agentname	Yes	The name of the agent who performed the identification.	HMUELLER
identificationtime	Yes	The time the identification was finished in ISO 8601 format.	2014-06-02T05:03:54Z
type	Yes	The channel used by the user. Can either be “WEB” or “APP”	WEB
transactionnumber	Yes	The transaction number passed by you when starting the request.	Your own internal ID (e.g. 287492_23552)
companyid	Yes	Your company id given to you during setup.	ihrebank
id	Yes	IDnow’s internal unique identifier for the identification.	IBA-H6GT2
filename	Yes	The name of the zip file containing detailed information about a single identification.	AH73JK3LM.zip
href	Yes	An absolute URL pointing to the location at which you can request the zip file from the REST interface with detailed information.	/api/v1/ihrebank/identifications/AH73JK3LM.zip

The result can have the following values.

Value	Type	Description
SUCCESS	Final	The identification has been performed without problems or changed data.
SUCCESS_DATA_CHANGE	Final	The identification has been performed without problems, but data has been changed compared to the initially provided data. This might happen if the user made a typo ("22.05.1982" instead of "22.05.1983"). The changed field will be marked with status "CHANGE" (see below).
FRAUD_SUSPICION_CONFIRMED	Final	The agent has a suspicion that the user tried to commit fraud. Additional details are available from IDnow on request.

In case you have activated the option to receive preliminary results in order to process the results in real-time, the status can also be one of the following values:

Value	Type	Description
REVIEW_PENDING	Preliminary	The identification has been finished successfully and the result is now waiting for review. The result is expected to be SUCCESS or SUCCESS_DATA_CHANGE after the review. For a realtime process this result can be treated as a successful identification.
FRAUD_SUSPICION_PENDING	Preliminary	The identification has finished but a fraud suspicion arised during the identification. The result is expected to be FRAUD_SUSPICION_CONFIRMED after the review. For a realtime process this result should be treated as a pending or failed identification.

13.1.2 Section Custom Data

This section holds the custom data which was passed when starting the identification. This section is marked by the key "customdata".

Parameter	Mandatory	Description	Example
custom1	No	Custom text field. Use this to pass your own IDs, tags etc.	Your own internal ID (e.g. 287492_23552)
custom2	No	See explanation for field custom1.	

custom3	No	See explanation for field custom1.	
custom4	No	See explanation for field custom1.	
custom5	No	See explanation for field custom1.	

13.1.3 Section Contact Data

This section holds the contact data of the user. This data is either passed when starting the identification, or entered by the user during the process. If the user changed data (for example the mobile phone) during the identification process, you will get back the updated data. This section is marked with the key “contactdata”.

Parameter	Mandatory	Description	Example
email	No	The user's email address.	sampleuser@example.com
mobilephone	No	The user's mobile phone number.	0151 23411232

13.1.4 Section User Data

This section holds the personal data of the user as retrieved during the identification. You will also get information if data has been changed. This section is marked by the key “userdata”.

Parameter	Mandatory	Description	Example
birthday	No	The users birthday in ISO 8601 format: YYYY-MM-DD	1975-12-20
birthname	No	The user's birthname. Does not include prefixes like “Geb.” Or “Geborene”.	MEIER
birthplace	No	The user's birthplace. All uppercase.	MÜNCHEN
city	No	The user's city. Will be provided in sub-object named “address”. All uppercase.	BERLIN
country	No	The user's country. Uppercase two-letter code as defined in ISO 3166. Will be provided in sub-object named “address”.	DE
firstname	Yes	The user's first name(s). All uppercase.	MICHAEL PETER
gender	No	The user's gender. Either 'MALE' or 'FEMALE'.	MALE
lastname	Yes	The user's last name. All uppercase.	BERGER
nationality	No	The user's nationality. Uppercase two-letter code as defined in ISO 3166.	DE

street	No	The user's street. Will be provided in sub-object named "address". All uppercase.	BAHNSTRASSE
streetnumber	No	The user's street number. This field can be configured to be part of the field "street", if you have street and number saved in one field in your database. If you wish to activate this setting please contact your technical account manager at IDnow.	27
title	No	Academic title. This will only be used, if the title is part of the name and shown in ID documents.	Dr.
zipcode	No	The user's zip code. Will be provided in sub-object named "address".	80127

All fields in this section can have additional data about the status of the field and about the original data if data has been changed. For details, please see the JSON and XML examples.

The status can have the following values.

Status	Description	Example
MATCH	The data retrieved from the identification document matches the data provided initially.	Original Data: "ARMIN" Data from ID: "ARMIN"
CHANGE	The data retrieved from the identification document is different from the data provided initially. Examples are typos by the user while opening a bank account. If a field is marked with "CHANGE", the result of the identification will always be "SUCCESS_DATA_CHANGED". Also, if a field is marked with "CHANGE", the original data is provided in "originaldata".	Original Data: "1982-05-22" Data from ID: "1983-05-22"
REDACTION	The data has been redacted, i.e. blackened out during the identification process and will not be provided through the API. This is a per field setting you can change during account setup.	
NEW	This data field has not been provided to IDnow. IDnow was able to retrieve it during the identification process. Therefore, no check between original data and identification data has taken place.	Original Data: <EMPTY> Data from ID: "1983-05-22"

Example 1: Matching firstname field in XML

```
...
<firstname status="MATCH">ARMIN</firstname>
...
```

Example 2: New nationality field in JSON

```
...
nationality: {
  value: "DE",
  status: "NEW"
},
...
```

Example 3: Changed birthday field in XML

```
...
<birthday status="CHANGE" original="1982-05-22">1983-05-22</birthday>
...
```

Example 4: Changed birthday field in JSON

```
...
birthday: {
  value: "1983-05-22",
  status: "CHANGED",
  original: "1982-05-22"
},
...
```

13.1.5 Section Identification Document

This section provides details about the identification document used by the user. This section is marked by the key "identificationdocument".

Parameter	Mandatory	Description	Example
type	Yes	The type of the identification document used by the user. Possible values are: <ul style="list-style-type: none"> IDCARD for ID card PASSPORT for passport DRIVERS_LICENSE for driver's license RESIDENCE_TITLE for residence title ("Aufenthaltsgenehmigung") 	IDCARD
country	Yes	The issuing country of the ID. Uppercase two-letter code as defined in ISO 3166.	DE
validuntil	Yes	The date until when the ID is valid in ISO 8601 format: YYYY-MM-DD.	2020-03-10
number	No	The ID number.	402324847

issuedby	No	The government agency who issued the ID. All uppercase.	LANDESHAUPTSTADT MÜNCHEN, KVR
dateissued	Yes	The data when the ID was issued in ISO 8601 format: YYYY-MM-DD.	2010-03-10
driversclasses	No	The driver's license classes present on the document as an array (only applicable for drivers licenses, details see below)	[{"type": "ML"}, {"type": "B"}]

Driver's License Classes (JSON example)

```
...
driversclasses: {
  value: [{
    type: "ML"
  }, {
    type: "B"
  }],
  status: "NEW"
},
...
```

13.1.6 Section Attachments

This section provides details about the additional attachments (audio logs, images, etc). This section is marked by the key "attachments".

Parameter	Mandatory	Description
idfrontside	Yes	The image filename showing the frontside of the ID. The filename is "<transactionnumber>_idfrontside.png" by default. For additional details see below.
idbackside	Yes	The image filename showing the backside of the ID. The filename is "<transactionnumber>_idbackside.png" by default. For additional details see below.
idholograms	No	The image filename showing the holograms of the ID. The filename is "<transactionnumber>_idholograms.png" by default.
userface	No	The image showing the face of the user. The filename is "<transactionnumber>_userface.png" by default. For additional details see below.
audiolog	No	The audiolog of the identification process. The filename is "<transactionnumber>_audiolog.wav" by default. For additional details see below.
<custom image key>	No	Additional images taken during identification process if any.

13.1.7 Section Questions

This section holds the answers to the questions which have either been pre-defined or answered by the identification agent. This section is marked by the key “questions”.

Parameter	Mandatory	Description	Example
<question key>	No	Key is the question key of the configured question. Value is the answer.	

The type of the value depends on the type of the question:

Question Type	Description	Value
RADIO_BOOLEAN	Input which allows to answers (“Yes / No”, “True / False”)	Stored as integer (0 = false, 1 = true)
RADIO_STRING	Multi-selection shown as radio buttons	String
DROPDOWN	Multi-selection shown as dropdown	String
DROPDOWN_COUNTRY	Country selection dropdown	String, uppercase two-letter code as defined in ISO 3166
INPUT	Text input field	String
NUMERIC	Numeric input field	Integer
DATE	Date picker	The users birthday in ISO 8601 format: YYYY-MM-DD
CUSTOM_IMAGE	An additional image taken during the process. The image will be stored with the question key in the filename.	True = Image is available, False = Image not available (0 = false, 1 = true). The image itself is stored in the zip, and can be seen in the attachments section as well.

13.2 Result Attachments

13.2.1 Image Format

The images provided in the identification can be provided in the following formats:

- PNG (default): The image in “Portable Network Graphics” with 24-bit palette.
- JPEG: The image in the JPEG format. The file ending will be “.jpg”

If you need a different format than PNG, please contact IDnow.

13.2.2 Audio Log

The audio log can be provided in the following formats:

- MP3 (default): Audio log encoded as mono, 22.05 kHz, 36 kbps MP3
- WAV: Audio log encoded as 11.025 kHz, 16 bit, mono

If you need a different format than WAV or MP3, please contact IDnow.

13.3 XML Result

The XML result will always start with the top-level “identifications” objects with one or more “identification” children.

```
<?xml version="1.0" encoding="UTF-8"?>
<identifications>
  <identification>
    <identificationprocess>
      <result>SUCCESS</result>
      <agentname>HMUELLER</agentname>
      <identificationtime>2014-06-
02T05:03:54Z</identificationtime>
      <type>WEB</type>
      <transactionnumber>AH73JK3LM</transactionnumber>
      <companyid>ihrebank</companyid>
      <id>IBA-H7GB6</id>
    </identificationprocess>
    <customdata>
      <custom1>2740332</custom1>
      <custom2>ABCD</custom2>
    </customdata>
    <contactdata>
      <email>muster@idnow.de</email>
      <mobilephone>+49176102030123</mobilephone>
    </contactdata>
    <userdata>
      <firstname status="MATCH">ARMIN</firstname>
      <lastname status="MATCH">BAUER</lastname>
      <birthday status="CHANGE" original="1982-05-
22">1983-05-22</birthday>
      <birthplace status="MATCH">MÜNCHEN</birthplace>
      <nationality status="MATCH">DE</nationality>
      <gender status="MATCH">MALE</gender>
      <address>
        <street status="MATCH">UNERTLSTR.</street>
        <streetnumber
status="MATCH">40</streetnumber>
        <city status="MATCH">MÜNCHEN</city>
        <country status="MATCH">DE</country>
        <zipcode status="MATCH">80469</zipcode>
      </address>
    </userdata>
    <identificationdocument>
      <type status="NEW">IDCARD</type>
      <country status="NEW">DE</country>
      <validuntil status="REDACTION"/>
      <number status="REDACTION"/>
      <issuedby status="NEW">LANDESHAUPTSTADT MÜNCHEN,
KREISVERWALTUNGSREFERAT</issuedby>
      <dateissued status="NEW">2012-03-27</dateissued>
    </identificationdocument>
    <attachments>
      <audiolog>AH73JK3LM_audiolog.wav</audiolog>
```

```

        <idfrontside>AH73JK3LM_idfrontside.png</idfrontside>
        <idbackside>AH73JK3LM_idbackside.png</idbackside>
        <userface>AH73JK3LM_userface.png</userface>
        <customimage>AH73JK3LM_customimage.png</customimage>
    </attachments>
    <questions>
        <question_key1>1</question_key1>
        <question_key2>value</question_key2>
        <question_key3>1983-05-22</question_key3>
    </questions>
</identification>
</identifications>

```

13.3.1 XML Result Signature

On request, IDnow can provide a signature for the XML file to prove that a) the XML file has not been altered and b) that the XML file is really coming from IDnow.

In the ZIP file the signature has the filename <transactionnumber>.sig.

The public keys can be received from the following locations:

Environment	Description	SHA1(.pem)
Test	https://go.idnow.de/assets/certs/idnow_signing_test_20171004.pem	cf85910dc4dd95f0fc8605b2ad39041a305b994a
Live	https://go.idnow.de/assets/certs/idnow_signing_20171018.pem	a970f6daa84014c12422217ef216c03b046322ab

The command to verify the signature of a XML file:

```

> openssl dgst -sha512 -verify <file>.pem -signature <transactionnumber>.sig
<transactionnumber>.xml
Verified OK

```

13.4 JSON Format

The JSON result will always start with the top-level “identifications” objects an array of one or more identification children.

```

{
  "identifications": [{
    "identificationprocess": {
      "result": "SUCCESS",
      "agentname": "HMUELLER",
      "identificationtime": "2014-06-02T05:03:54Z",
      "type": "WEB",
      "transactionnumber": "AH73JK3LM",
      "companyid": "ihrebank",
      "id": "IBA-H7GB6"
    },
    "customdata": {

```

```
        "custom1": "2740332",
        "custom2": "ABCD"
    },
    "contactdata": {
        "email": "muster@idnow.de",
        "mobilephone": "+49176102030123"
    },
    "userdata": {
        "firstname": {
            "value": "ARMIN",
            "status": "MATCH"
        },
        "lastname": {
            "value": "BAUER",
            "status": "MATCH"
        },
        "birthday": {
            "value": "1983-05-22",
            "status": "CHANGE",
            "original": "1982-05-22"
        },
        "birthplace": {
            "value": "MÜNCHEN",
            "status": "MATCH"
        },
        "nationality": {
            "value": "DE",
            "status": "MATCH"
        },
        "gender": {
            "value": "MALE",
            "status": "MATCH"
        },
        "address": {
            "street": {
                "value": "UNERTLSTR.",
                "status": "MATCH"
            },
            "streetnumber": {
                "value": "40",
                "status": "MATCH"
            },
            "city": {
                "value": "MÜNCHEN",
                "status": "MATCH"
            },
            "zipcode": {
                "value": "80469",
                "status": "MATCH"
            },
            "country": {
                "value": "DE",
                "status": "MATCH"
            }
        }
    },
    "identificationdocument": {
        "type": {
```

```
        "value": "IDCARD",
        "status": "NEW"
    },
    "country": {
        "value": "DE",
        "status": "NEW"
    },
    "validuntil": {
        "value": "2020-03-10",
        "status": "NEW"
    },
    "number": {
        "value": null,
        "status": "REDACTION"
    },
    "issuedby": {
        "value": "LANDESHAUPTSTADTMÜNCHEN, KREISVERWALTUNGSREFERAT",
        "status": "NEW"
    },
    "dateissued": {
        "value": "2012-03-27",
        "status": "NEW"
    }
},
"attachments": {
    "audiolog": "AH73JK3LM_audiolog.wav",
    "idfrontside": "AH73JK3LM_idfrontside.jpg",
    "idbackside": "AH73JK3LM_idbackside.jpg",
    "userface": "AH73JK3LM_userface.jpg",
    "customimage": "AH73JK3LM_customimage.jpg"
},
"questions": {
    "question_key1": {
        "value": 1
    },
    "question_key2": {
        "value": "value"
    },
    "question_key3": {
        "value": "2012-03-27"
    }
}
}
}]
}
```

14 Webhooks

14.1 Summary

When this feature is activated for your account, after each identification a backend call (JSON POST) to a URL on your servers will be executed by the IDnow system.

To activate the feature please contact your technical account manager at IDnow.

14.2 Timing

IDnow can execute different webhooks depending on your needs

Name	Timing of calls
REALTIME	After the identification has been finished (preliminary result).
FINAL	After the review has been finished (final result). The realtime and the final webhook can be distinguished by reading the field "result" from the result data (see "13.1 Data Fields").
ABORTED	If the agent aborted the identification. Example reasons are technical errors like the user's camera or audio is not working. You will also get the reason why the call failed.
CANCELED	If the agent detects invalid data during the review that cannot be corrected. E.g. the reviewer decides that the image quality was not good enough and cancels the identification after the realtime result has already been sent. While this can happen, its rather rare (about 0.1 % of the identifications)

An example:

- Your account has realtime results active.
- User tries to do an identification but fails since his internet connection is not good enough. You will receive a webhook for this aborted identification.
- User tries again and successfully finishes the identification. You receive a realtime webhook right after the identification with status REVIEW_PENDING.
- IDnow finishes the review. You receive another webhook with the final results.

14.3 IPs

Environment	Webhook-IPs
Live	62.128.13.228 62.128.13.229
Test	52.30.27.5 52.48.216.0

14.4 Realtime / Final Webhooks

14.4.1 JSON Content

The JSON of one identification is passed in the body of the call to the web hook.

```
{
  "identificationprocess": {
    ...rest of JSON...
  },
  "customdata": {
    ...rest of JSON...
  },
  ...rest of JSON...
}
```

The content of the single identification result is the same as the JSON format from the last chapter.

14.4.2 Example for Typical Usage

IDnow informs you about a new identification via the web hook call, so that you can automatically trigger a SFTP download of the identification's result.

14.5 Aborted / Canceled Webhooks

The JSON structure is similar to the JSON from success web hooks.

The differences are:

JSON attribute	Description
identificationprocess.result	Values are "ABORTED" - identification was aborted during identification "CANCELED" - identification was canceled during review process
identificationprocess.reason	Reason why the identification was aborted / canceled. Values are defined and described in the section "Reasons for failure".
userdata.key.value	The value of the userdata is not available (since the identification failed). Only the original value is available.
attachments	Attachments section is not available.
Identificationdocument	Identification document section is not available.

14.5.1 JSON Content

The JSON of one identification is passed in the body of the call to the web hook.

```
{
  "identificationprocess": {
    "result": "CANCELED",
    "reason": "DATA_APPLICATION_DATA",
  }
}
```

```

    "companyid": "testbank",
    "agentname": "HMUELLER",
    "identificationtime": "2015-10-20T17:40:37+02:00",
    "id": "XKH-UNGNP",
    "type": "WEB",
    "transactionnumber": "IDN-x-83176070"
  },
  "customdata": {
    ... rest of JSON ...
  },
  "contactdata": {
    ... rest of JSON ...
  },
  "userdata": {
    "birthday": {
      "status": "ORIGINAL",
      "original": "2005-10-13"
      ... attribute "value" not included ...
    }
    ... rest of JSON ...
  }
  ... attachments section not included ...
  ... identificationdocument section not included ...
}

```

14.5.2 Example for Typical Usage

IDnow informs you about failed and canceled identification so that you can track the progress of the user and inform the user that he should redo the identification.

14.5.3 Reasons for Failure

Reason	Description	Perma- nent?	Caused by User?
DATA_APPLICATION_ADDRESS	Address from application does not match address from id document and cannot be corrected.	yes	yes
DATA_APPLICATION_DATA	Data from application and data from the id document do not match and cannot be corrected.	yes	yes
DATA_ID_EXPIRED	The validity date of the id document has expired. User must just a different document.	no	yes
OTHER_ABUSE	Abuse of the procedure. Inappropriate behavior.	no	yes
OTHER_MISCELLANEOUS_PERMANENT	Other error, which is not covered by specific error. Cannot be solved.	yes	
OTHER_MISCELLANEOUS_TEMPORARY	Other error, which is not covered by specific error. Can be solved by a retry.	no	
OTHER_TEST	Identification was a test call.	no	yes
TECH_AUDIO	Poor audio quality. Audio quality is poor or user / agent inaudible.	no	yes

TECH_DISCONNECTED_VIDEO	The video stream has disconnected during the identification.	no	yes
TECH_DISCONNECTED_WEBSOCKET	The websocket connection is disconnected during the identification. Normally caused by loss of internet connection by the user.	no	yes
TECH_HOLOGRAM	Security features not visible or broken. No suspicion of fraud.	no	yes
TECH_ID_TYPE	Id document is not allowed or is not supported.	no	yes
TECH_IDENT_CODE_DELIVERY	User was unable to receive the ident code SMS.	no	yes
TECH_INTERNAL_SERVER_ERROR	An error occurred on server.	no	no
TECH_INTERNET_CONNECTION	Internet connection of the user is not fast enough.	no	yes
TECH_LIGHTING	Poor lighting. Person or id document not sufficiently visible.	no	yes
TECH_PHOTO	Poor photo quality. Person or id document not sufficiently sharp and recognizable.	no	yes
TECH_TIMEOUT	The identification request was open too long. Identification requests get aborted after 30 minutes.	no	yes
TECH_VIDEO	Poor video quality or camera not good enough. Person or id document not sufficiently sharp and recognizable.	no	yes
USER_ABORT_WHILE_WAITING	User has aborted the identification while he was waiting for an available agent.	no	yes
USER_CANCELLATION	User has aborted the identification.	no	yes
USER_ID_NUMBER	User repeatedly reads the wrong id document number.	no	yes
USER_IDENT_CODE	Ident code repeatedly entered incorrectly.	no	yes
USER_LANGUAGE	User speaks unsupported language.	yes	yes
USER_NO_ID	User has currently no id card available.	no	yes
USER_WRONG_PERSON	The person of application and identification do not match. E.g. Husband has opened bank account, but wife is in the identification call. No suspicion of fraud.	no	yes

15 Testing

To test whether your application is correctly communicating with the IDnow application, IDnow supports 3 different methods of processing an identification. Depending on the type of test, different parts of the procedure are automated on the side of IDnow or the company, respectively:

Type	User	Agent
Automated	Company Test Implementation	IDnow Test-Robot
Manual	Human	IDnow Test-Robot
With Agent	Human	Human

Where “IDnow Test-Robot” is a service running on the test environment and simulates a call center agent by mindlessly clicking through the procedure. The term “Company Test Implementation” refers to a custom implementation of the company to automate the testing on their side. For details continue reading.

15.1 Selecting a Test Scenario

To select a testscenario you can either set the firstname, lastname or one of the custom fields to a special setting “<Prefix>-<test scenario>”. The prefix determines which type of test is performed. Available prefixes are:

Prefix	Description
X-MANUALTEST	Performs a test where you can use the web or app, but the agent is automated.
X-AUTOTEST	Both the user and the agent are automated. No user interaction required.

The available test scenarios are:

Testcase	Description
HAPPYPATH	Perform a happy path test. Ident is finished successfully and no changes are made.
CHANGEALL	Performs a successful identification, but changes all fields to new values. In addition, all allowed UTF-8 characters supported by IDnow are returned.
CHANGEALLREVIEW	Performs a successful identification, but changes all fields during review (and not during the identification like “CHANGEALL”).
ABORTIDENT	The agent aborts during the identification (e.g. the video quality is not good enough).
FRAUDIDENT	The agent reports fraud suspicion during the identification and the fraud is confirmed in the review.
FRAUDREVIEW	The agent performs a successful identification, but during review a fraud is detected and confirmed.
FRAUDOK	The agent reports fraud suspicion during the identification but during the review the identification is marked as legit.

CANCELLED	The agent performs a successful identification, but during the review it is detected that the ident was not performed correctly (e.g. the picture quality is not good enough).
LONGREVIEW	Normally the review of the test scenarios is performed right away (~1-2 Minutes delay). Using the LONGREVIEW scenario the review is performed 24 hours later.
HOLDCERTIFICATE	The agent performs a successful identification, but the system sends the out the signed documents and the results after 10 minutes. (this test is only performable with an eSigning identification)

15.2 Automated Tests

To perform an automated test, first create a new identification using the REST API (see chapter 3). Please select one of the test scenarios from above and set the firstname, lastname or custom field accordingly.

The prefix to use is “X-AUTOTEST”. As an example for performing a happy path test:

```
firstname: "X-AUTOTEST-HAPPYPATH"
```

Once everything is set up (identification created, identification started), the following POST starts the test. Note that using the Test-Robots requires using the api subdomain. Also see the following example.

15.2.1 Path

After creating a unique transaction token for the identification POST to:

```
/api/<version>/<companyid>/identifications/<transactionnumber>/requestVideoChat
```

Example: /api/v1/company-xyz/identifications/1234567890/requestVideoChat

15.2.2 Header

Field	Mandatory	Content	Description
Content-Type	Yes	application/json	The mime-type

15.2.3 Body

The body is an empty JSON node: “{}”.

15.2.4 Example

This example assumes that you are using the following settings:

companyid	ihrebank
transactionnumber	1234567890
apiKey	exampleApiKey

- 1) Create a new identification (see chapter 3.2.11) with one of the test scenarios in the data.
- 2) If testing with an eSigning identification, upload the document (see chapter 7.2.4.2).
- 3) Start the ident with:

```
curl -i --header "X-API-KEY: exmapleApiKey " --header "Content-Type: application/json" -d "{}" https://api.test.idnow.de/api/v1/ihrebank/identifications/1234567890/start
```

- 4) Request the video chat

```
curl -i --header "Content-Type: application/json" -d "{}" https://api.test.idnow.de/api/v1/ihrebank/identifications/1234567890/requestVideoChat
```

If called with valid credentials, the server will respond:

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Content-Length: 2

{}
```

15.3 Manual Test

A manual test means that you can use the frontend or apps of IDnow yourself, but the agent part is automated. For this, create a new identification using the REST API (details see above) or using the userdata formular.

The prefix to use is "X-MANUALTEST". As an example for performing a fraud test:

```
firstname: "X-MANUALTEST-FRAUDIDENT"
```

One possible alternative would be: setting lastname in the userdata webform to "X-MANUALTEST-FRAUDIDENT".

15.4 Test with an IDnow Agent

You can request a test with an IDnow agent on the test environment. Please contact acm@idnow.de for a testing time slot. An IDnow agent will be available for you on the test environment. Please note: Since we have to use our agents for this form of testing we have to charge hourly rates for this sort of testing.

15.5 Checking the Results

Using one of the methods mentioned for retrieving data, your application can process the data resulting from the identification process on the test environment.