

WebServices

identity Trust Management AG

Customer Services



API Reference
Version 2.09b
2020-05-20

Index

1	Changelog	4
2	General Information	5
2.1	Contact and Credentials for Service Integration	5
3	Definition of Services	5
4	Before Creating an Order	6
4.1	Checking availability of designated video agents	6
4.2	Checking sign-me user status	6
4.2.1	checkSignmeUser Requirement	6
4.2.2	checkSignmeUser URL	6
4.2.3	checkSignmeUser Post Data	6
4.2.4	checkSignmeUser Response Data	7
4.2.5	checkSignmeUser HTTP Codes	7
4.2.6	checkSignmeUser Example Request	7
4.2.7	checkSignmeUser Example Response	7
5	Creating orders	8
5.1	putOrder	8
5.1.1	putOrder Requirements	8
5.1.2	putOrder URL	8
5.1.3	putOrder PUT Data	8
5.1.4	putOrder Response Data	12
5.1.5	putOrder HTTP Codes	12
5.1.6	putOrder Example Request	13
5.1.7	putOrder Example Response	14
6	After an Order was completed	15
6.1	Redirects	15
6.2	WebHooks	15
6.3	getStatus	16
6.3.1	getStatus URL	16
6.3.2	getStatus Response Data	17
6.3.3	getStatus HTTP Codes	18
6.3.4	getStatus Example Request	19
6.3.5	getStatus Example Response	19
6.4	getIdentData	20
6.4.1	getIdentData Requirements	20
6.4.2	getIdentData URL	20
6.4.3	getIdentData variants	21
6.4.4	getIdentData Response Data	21
6.4.5	getIdentData HTTP Codes	22
6.4.6	getIdentData Example Request	23
6.4.7	getIdentData Example Response	23
6.5	getIdentDataPDF	24
6.6	getIdentDataPDF Requirement	24
6.6.1	getIdentDataPDF URL	24
6.6.2	getIdentDataPDF HTTP Codes	24
6.6.3	getIdentDataPDF Example Request	25
6.6.4	getIdentDataPDF Example Response	25
6.7	getESignPDF	26

6.8	getESignPDF Requirement	26
6.8.1	getESignPDF URL	26
6.8.2	getESignPDF Response Data	26
6.8.3	getESignPDF HTTP Codes	26
6.9	getESignHash	27
6.9.1	getESignHash Requirement	27
6.9.2	getESignHash URL	27
6.9.3	getESignHash Response Data	27
6.9.4	getIdentData HTTP Codes	27
6.10	getVideoFileBinary	28
6.10.1	getVideoFileBinary URL Requirement	28
6.10.2	getVideoFileBinary URL	28
6.10.3	getVideoFileBinary HTTP Codes	28
6.11	getVoiceFiles	29
6.11.1	getVoiceFiles URL	29
6.11.2	getVoiceFiles HTTP Codes	29
6.11.3	getVoiceFiles Example Request	29
6.11.4	getVoiceFiles Example Response	29
6.12	delIdentdata	30
6.12.1	delIdentdata URL	30
6.12.2	delIdentdata HTTP Codes	30
7	How to handle incomplete Orders	31
7.1	Setting up a TimeOut	31
7.2	getAllStatus	31
7.2.1	getAllStatus URL	31
7.2.2	getAllStatus POST Data (optional)	31
7.2.3	getAllStatus Response Data	31
7.2.4	getAllStatus HTTP Codes	32
7.2.5	getAllStatus Request Example	33
7.2.6	getAllStatus Response Example	33
7.3	cancelOrder	34
7.3.1	cancelOrder URL	34
7.3.2	cancelOrder Example URL	34
7.3.3	cancelOrder Response Data	34
8	Re-using identifications to sign documents	35
9	Monitoring	36
9.1	serverStatus	36
10	IP-Ranges currently used	37
11	Technical requirements for Video Ident	38

1 Changelog

Version	Date	Author	Comment
2.0	2017-03-06	Markus Herzog	initial new Version
	2017-03-16	Markus Herzog	New Method getVoiceFiles
2.01	2017-04-26	Markus Herzog	New Method getVideoQueueStatus Added Status 89 to ExtendedStatus List
2.02	2017-05-17	Markus Herzog	Added Object RedirectURL on putOrder Changed putOrder.Add possible Options
2.03	2017-05-30	Markus Herzog	New Method getShops
2.031	2017-07-31	Markus Herzog	New Method getVideoFileBinary Added Object WebhookURL on putOrder
2.032	2017-08-18	Markus Herzog	New Method getESignHash
2.032b	2017-11-20	Markus Herzog	Added new Type for Additional Field Country ISO on putOrder
2.032c	2018-02-12	Markus Herzog	Error in Definition getIdentData->IDCard->Type corrected
2.032d	2018-03-20	Markus Herzog	Added Object eSignHash on putOrder
2.04	2018-05-28	Markus Herzog	New Method checkSignmeUser
2.05		Markus Herzog	Added Product AutoID
2.06	2018-10-04	Erkan Ciftci	Removed endpoints: getVideoTargetLink, delRecheck, getVideoQueueStatus and getESignLink; Removed putOrder elements: Lang, ReCheck; Added several descriptions and sign-me features
2.06 b	2018-11-29	Erkan Ciftci	Minor corrections and clarifications
2.07	2018-10-09	Markus Herzog	new Endpoint, added Errorcodes on some errors
2.08	2019-01-04	Markus Herzog	Internal Processing changes
2.09	2019-01-21	Markus Herzog	Added Element AllowedProductSet to putOrder Added Field DateOfIssue to IdentData->IDCard
2.09a	2019-03-01	Erkan Ciftci	Approved Version
2.09b	2020-05-20	Timo Tegtmeier	Minor clarifications and extensions

2 General Information

2.1 Contact and Credentials for Service Integration

Throughout the technical implementation you are being assisted by identity Trust Management AG. itsupport@identity.tm is the contact for any kind of technical and procedural questions. If you have not already received backend credentials for the services in this document please apply for some by contacting identity.

3 Definition of Services

Service Kind

RESTful Webservice by HTTP (http 1.1 Status Codes) - JSON Data in HTTP Body

Endpoint

<https://customer.identity.tm/api/2.09/>

Content-Type

application/json

Authentication

BasicAuth with Username (CustomerID) and Password (CustomerCode)

Types in Descriptions

O= Object, A=Array, S=String, N=Number, B=Boolean

Field Types

Date: YYYY-MM-DD

Date Time: ISO 8601 - Example: 2013-07-15T18:00:00+02:00

MSISDN: International Phonenumber without blanks. Example: +491701234567

Country: The ISO-Code of the country (ISO 3166) e.g. DE, AT, NL

TLS

Version 1.2

4 Before Creating an Order

4.1 Checking availability of designated video agents

In case you are creating orders for video identifications and you wish to check prior how the availabilities of the designated agents are in a given moment, you can apply for a separate endpoint by contacting itsupport@identity.tm. This service may make sense if you are using different identity providers and would like to implement a load balancer between those.

4.2 Checking sign-me user status

This method is only applicable, if you wish to create an eSign order for sign-me (D-Trust). Before creating the order, you can check prior the user status in the sign-me system in order to find out, if the given user already has a sign-me account and if he is able to sign with a qualified electronic signature (QES). Both elements in the response need to be true in order to create a plain signature Order (Add:16 in putOrder). Otherwise the account needs to be created and verified (Add:0 or 8 in putOrder – The Add value 0 represents the default setting with creation/verification of the account plus PDF document(s) that need(s) to be signed – With the Add value 8 it is the same with the exception of the PDF and signature part).

4.2.1 checkSignmeUser Requirement

Your account must be enabled for eSign and sign-me by identity.

4.2.2 checkSignmeUser URL

[POST] <https://customer.identity.tm/api/2.09/checkSignmeUser>

4.2.3 checkSignmeUser Post Data

Element	Superior element	Data type	Length	Occurs	Description
Email	[ROOT]	String		Yes	The SignMe Username (Email)

4.2.4 checkSignmeUser Response Data

Element	Superior element	Data type	Length	Occurs	Description
UserExists	[ROOT]	Bool			The Signme Account exists
UserIsAble ToSign	[ROOT]	Bool			The Signme Account is qualified

4.2.5 checkSignmeUser HTTP Codes

200 plus JSON document

401 if customer authentication fails, please check username and password

400 other error: JSON document with error in body

4.2.6 checkSignmeUser Example Request

```
{
  "Email": "e.mustermann@example.com"
}
```

4.2.7 checkSignmeUser Example Response

```
{
  "UserExists": false,
  "UserIsAbleToSign": false
}
```

5 Creating orders

5.1 putOrder

This central method is designated to create any kind of new order for identifications and/or qualified electronic signatures to natural persons.

5.1.1 putOrder Requirements

The desired products must be associated with and enabled on your account by identity.

5.1.2 putOrder URL

[PUT] <https://customer.identity.tm/api/2.09/putOrder>

5.1.3 putOrder PUT Data

Element	Superior element	Data type	Length	Mandatory	Description
Ref	[ROOT]	String	100	No	Your internal number to the order. This field has no meaning for identity and can be used to associate an order with an ID on your systems.
Product	[ROOT]	Int		Yes	Specifies the initial product. 0 = All enabled products <u>except</u> Video + eSign + AutoID 12 = All enabled products <u>including</u> Video, but <u>except</u> eSign + autoID 15 = eSign 16 = autoID (Beta)
Add	[ROOT]	Int		Yes	Specifies certain additional product behaviors. 0 will be the correct choice in most cases. 0 = normal behavior 8 = if sign-me: generate account 16 = if sign-me: use exist. account

					32 = no acquisition by phone 4096 = License ID Check 8192 = License Check (without ID) 131072 = Record Video 268435456 = PickIdent
AllowedProducts	[ROOT]	Array		No	Array of Product IDs the order is allowed to be changed to
getTargetLink	[ROOT]	bool		No	to get TargetURL in Response
getVideoHashes	[ROOT]	bool		No	if Product 12 or 15
RedirectURL	[ROOT]	Object		No	Individual Redirect URLs for Video Sessions will be used if set. [Ref], [OrderID] as Placeholder
Success	RedirectURL	String	150		
Abort	RedirectURL	String	150		
WebhookURL	[ROOT]	Object		No	[OrderID] as Placeholder
AfterVideo	WebhookURL	String	150	No	
OnFinal	WebhookURL	String	150	No	
Phone	[ROOT]	String	50	No	Phone number of your customer
Mobile	[ROOT]	String	50	No	Mobile number of your customer MSISDN Format
Email	[ROOT]	String	50	Yes	Email address of your customer
Address	[ROOT]	Object		Yes	Mandatory if your customer is allowed to Products Courier, Shop
Firstname	Address	String	50	Yes	Physical Address data of your customer (Contact / Delivery)
Surname	Address	String	50	Yes	
CompanyName	Address	String	50	No	
Street	Address	String	50	No	
ZipCode	Address	String	5	No	
City	Address	String	100	No	
Country	Address	String	3	Yes	The ISO-Code of the country (ISO 3166) e.g. DE, AT, NL
IdentData	[ROOT]	Object		No	The fields you provide, will be checked. Empty fields will be picked if possible.
Firstname	IdentData	String	50	Depends	Please give us notice if you plan to omit any of these fields.
Surname	IdentData	String	50	Depends	
Street	IdentData	String	100	Depends	
ZipCode	IdentData	String	5	Depends	
City	IdentData	String	100	Depends	
Country	IdentData	String	2	Depends	

Birthdate	IdentData	Date		Depends	YYYY-MM-DD
Birthplace	IdentData	String	100	Depends	
Nationality	IdentData	String	50	Depends	
Title	IdentData	String	50	Depends	
Artistname	IdentData	String	50	Depends	
Gender	IdentData	String		Depends	female/male
IDCard	IdentData	Object		No	
CardNo	IDCard	String	50	Depends	
PlaceOfIssue	IDCard	String	100	Depends	
DateOfIssue	IDCard	Date		Depends	
ValidUntil	IDCard	Date		Depends	YYYY-MM-DD
DriverLicence	IdentData	Object		Depends	If DriverLicense Check with Add 4096 and/or 8192, provide data if available
LicenceNo	DriverLicence	String	50	Depends	
Classes	DriverLicence	String	50	Depends	
DateOfIssue	DriverLicence	Date		Depends	
Additional	IdentData	Object		No	If you need special attributes, you can add here for Courier/Shop up to 2, Video up to 10 items. <detail> can have a length of 50
<detail>	Additional	Object		No	
Type	<detail>	Int			1 = String, 2 = Date, 3 = Bool, 4 = Country ISO 3166
Label	<detail>	String	100		Readable Label for the Element
Value	<detail>	String	150		
LegitimationDocument	[ROOT]	Array		No	Individual Legitimation Documents, if default selection is not sufficient
Document	LegitimationDocument	String	50		The Readable Description shown to the Agent/Employee
eSignPDF	[ROOT]	Object/ Array		No	Document(s) designated for QES, array for multiple documents.
Data	eSignPDF	String		No	Base64 encoded PDF (PDF must be unprotected)
Filename	eSignPDF	String	50	No	Filename
SignaturePosX	eSignPDF	Int		No	Only applicable for sign-me. The units are PDF Points equivalent to 1/72 inch. The lower/left corner of the document is the origin X=0 and Y=0.
SignaturePosY	eSignPDF	Int		No	As above

SignatureHeight	eSignPDF	Int		No	Only applicable for sign-me and if SignaturePos is also set. The units are PDF Points equivalent to 1/72 inch.
SignatureWidth	eSignPDF	Int		No	As above
eSignHash	[ROOT]	Object/ Array		No	Only applicable for sign-me
Hash	eSignHash	String		No	Base64 Hash of prepared PDF
Filename	eSignHash	String	50	No	Filename
SignaturePosX	eSignHash	Int		No	Only applicable for sign-me. The units are PDF Points equivalent to 1/72 inch. The lower/left corner of the document is the origin X=0 and Y=0.
SignaturePosY	eSignHash	Int		No	As above
SignatureHeight	eSignHash	Int		No	Only applicable for sign-me and if SignaturePos is also set. The units are PDF Points equivalent to 1/72 inch.
SignatureWidth	eSignHash	Int		No	As above
eSignPDF	[ROOT]	Object		No	
Data	eSignPDF	String		No	Base64 encoded PDF (PDF must be unprotected)
Filename	eSignPDF	String	50	No	Filename
eSignHash	[ROOT]	Object		No	
Hash	eSignHash	String		No	Base64 Hash of prepared PDF
Filename	eSignHash	String	50	No	Filename
Info	eSignHash	String	50	No	Info to the PDF Document (e.g. what kind of contract)
eSignPartnerOrderID	[ROOT]	String	20	No	To reference an eSign Order to another. Contract from first Order will be signed by all parties.
IdentDataSigned	[ROOT]	String		No	Previously generated identification data for signing purposes

5.1.4 putOrder Response Data

Element	Superior element	Data type	Length	Occurs	Description
OrderID	[ROOT]	String	20	Depends	Our internal ID of the Order, always unique
Ref	[ROOT]	String	100	If given	Your internal ID of the Order, can be set multiple times
TargetURL	[ROOT]	String	100	Depends	URL to redirect your customer
VideoTargetLink	[ROOT]	String	100	Depends	Deprecated URL to redirect your customer directly into Video (if VideoHashes set to true)
VideoHashes	[ROOT]			Depends	
ShortCode	VideoHashes	String	5		Reference number to start Videoidentification in identity App on iOS/Android
LongCode	VideoHashes	String	20		VideoHash to authenticate your customer with your own App on iOS/Android (SDK)
Error	[ROOT]	Text		Depends	

5.1.5 putOrder HTTP Codes

202 plus JSON document

401 if customer authentication fails, please check username and password

400 other error: JSON document with error in body

5.1.6 putOrder Example Request

```
{
  "Ref": "Sign-me_Account_Creation_Test",
  "Product": 15,
  "Add": 8,
  "getTargetLink": true,
  "getVideoHashes": true,
  "WebhookURL": {
    "OnFinal": "https://webhook.site/1234567_890?OnFinal=[OrderID]",
  },
  "RedirectURL": {
    "Success": "https://www.example.com/redirect/positive?OrderID=[OrderID]",
    "Abort": "https://www.example.com/redirect/negative?OrderID=[OrderID]",
  },
  "Email": "e.mustermann@example.com ",
  "Mobile": "+491711234567",
  "Address": {
    "Firstname": "Erika",
    "Surname": "Mustermann",
    "Street": "Heidestr. 17",
    "ZipCode": "43000",
    "City": "Köln",
    "Country": "DE"
  },
},
  "IdentData": {
    "Firstname": "Erika",
    "Surname": "Mustermann",
    "Street": "Heidestr. 17",
    "ZipCode": "43000",
    "City": "Köln",
    "Birthdate": "1964-08-12",
    "Birthplace": "Berlin",
    "Additional": {
      "idCardDateOfIssue": {
        "Type": 2,
        "Label": "Date of Issue",
        "Value": "2010-01-11"
      }
    }
  }
}
```

5.1.7 putOrder Example Response

```
{
  "OrderID": "99921269855041",
  "Ref": "Sign-me_Account_Creation_Test",
  "TargetURL":
  "https://www.identity.tm/status_neu/37C4B60A72A1C6C74FC038282CED500B",
  "VideoTargetLink": "https://video.identity.tm/identification/a9PPz-
eFcNBifsk--bNJ",
  "VideoHashes": {
    "ShortCode": "SHU8T",
    "LongCode": "a9PPz-eFcNBifsk--bNJ"
  }
}
```

6 After an Order was completed

When an order has been completed, the user will be redirected to the specified URL and the according webhook will be called.

6.1 Redirects

By default, the user will be redirected to the designated URL when the identification is finished from an end-user perspective. Keep in mind, that this moment will most probably not be the moment in which the order is finally processed, so no result data might be available. You have to wait for the OnFinal webhook to be triggered, before retrieving results.

Alternatively your account can be configured to keep the user on a spinner page on identity.tm side until the order is completed. In this case, data will be available when the redirect happens. Please contact itsupport@identity.tm if you would like to have such a spinner or more information on this topic.

6.2 WebHooks

Like the Redirect URLs, the Webhook URLs can also be set up within each putOrder request (WebHook element) that is being made. A different approach would be to set it up globally for your accounts on our side. In both cases we differentiate Webhooks after a video call (AfterVideo) has been completed or cancelled and final Webhooks (OnFinal) when the order was completed (e.g. user has signed the documents) and all files are ready on our backend for collection. GET params and placeholders are [Ref] and/or [OrderID]. Ref is your internal reference number that is defined in the putOrder request by you. It is possible to create multiple orders with the same Ref value. The OrderID is our internal number which is always unique and defined by us in the putOrder response.

Examples:

```
https://www.example.com/identity/callback/AfterVideo?orderID=[OrderID]&ref=[Ref]  
https://www.example.com/identity/callback/OnFinal?orderID=[OrderID]&ref=[Ref]
```

6.3 getStatus

You can get all status codes for a the given order. The complete status codes are listed in chapter 6.3.2. Especially after an OnFinal webhook you may check what the final status of the given order is. Depending if QES is involved or not, the following status kinds are most importantly to be noted:

Final status kinds for identification order without QES

Status Code	Status Kind
6	Positive identification
17	Negative identification
22	Manually terminated
89	Fraud Suspicion
99	TimeOut

Final status kinds for identification order with QES

Status Code	Status Kind
84	Document(s) signed
17	Negative identification
22	Manually terminated
89	Fraud Suspicion
99	TimeOut
100	TimeOut after identification

6.3.1 getStatus URL

[GET] <https://customer.identity.tm/api/2.09/getStatus/OrderID/ExtendedList>

[GET] <https://customer.identity.tm/api/2.09/getStatus/OrderID>

6.3.2 getStatus Response Data

Bold written status codes are returned by default. Non-bold status codes are only included if ExtendedList is set.

Kind	Definition EN Long	Definition DE Short
3	Order was loaded by courier	Beladen
6	Identification positively conducted	Zustellung
7	Email was sent	EMail
9	Order on the way to the central HUB	Ausgang Depot
10	Order resides in the depot	Eingang Depot
12	Customer couldn't be reached by phone	TNE
14	Document sent to client	Retour Z
15	Customer wasn't present at appointment and was notified	Na
16	Identification refused by customer	Av
17	Identification negatively conducted	In
19	Appointment conducted with customer	Tv
20	Incorrect address in order	FA
21	The processing was aborted	Abbruch
22	The processing was terminated	Retour
23	Document verified	Vertrag OK
24	Document erroneous in postprocessing	Vertrag fehlerhaft
25	Order initialized	Daten erfasst
26	Follow-up scheduling	TV Wiedervorlage
33	Giro Forwarding	Giro Weiterleitung
34	Giro Feedback	Giro - Timeout
49	eID IDapp link created	eID Ausweisapp Link gebaut
50	eID IDapp request	eID Ausweisapp Request
51	eID data gathered	eID Daten erfasst
54	Support	Kundenanfrage bearbeitet
55	Customer couldn't be reached by phone	2xTNE
56	Phone number is incorrect	Datenkorrektur
57	Identification not possible, customer repeatedly wasn't present at appointments	AND
58	Identification refused by customer by phone	telefonischer AV
63	Cancellation	Storno
64	Appointment cancellation	TV Storno
66	Order is physically on the way to client	RetourZ Sendung

68	Change of procedure to Shop	Verfahrenswechsel zu Shop
69	Change of procedure to Home (Courier)	Verfahrenswechsel zu Home
70	Spoken on answering machine	Anrufbeantworter besprochen
71	SMS sent	SMS gesendet
73	Change of procedure to Video	Verfahrenswechsel zu Video
74	Change of procedure to eID	Verfahrenswechsel zu eID
75	Video identification conducted	Video Ident durchgeführt
77	Video identification in waiting room	Video Ident im Warteraum
78	Video identification call has begun	Video Ident Call begonnen
79	Video identification call has been left	Video Ident Call verlassen
80	Identification data ready for pick-up	Identdaten abholbereit
81	Change of procedure to Giro	Verfahrenswechsel zu Giro
83	Document signature request	Dokument SignaturRequest
84	Document signed	Dokument signiert
89	Fraud attempt by customer	Betrugsverdacht
94	eSign Account created	eSign Konto angelegt
95	eSign Account verified	eSign Konto verifiziert
96	eSign Account already exists	eSign Konto ist schon existent
97	Images were sent to Video	Images an Videosystem
98	Change of procedure to AutoID	Verfahrenswechsel zu AutoID
99	TimeOut	TimeOut
100	TimeOut after Ident	TimeOut nach Ident
101	eSign documents verified and hashes generated	eSign Dokumente bestätigt/Hash erzeugt
102	eSin document preview	eSign Dokumente Preview

6.3.3 getStatus HTTP Codes

200 plus JSON document

401 if customer authentication fails, please check username and password

400 other error: JSON document with error in body

404 if not found

6.3.4 getStatus Example Request

```
[GET] https://customer.identity.tm/api/2.0/getStatus/99921269855041/ExtendedList
```

6.3.5 getStatus Example Response

```
{
  "OrderID": "99921269855041",
  "Ref": "Sign-me_Account_Creation_Test",
  "Executed": "2018-09-05T11:06:21+02:00",
  "Status": [
    {
      "Kind": 25,
      "Time": "2018-09-05T10:53:59+02:00"
    },
    {
      "Kind": 71,
      "Time": "2018-09-05T10:54:00+02:00",
      "Text": "signme Passwort"
    },
    {
      "Kind": 94,
      "Time": "2018-09-05T10:54:00+02:00"
    },
    {
      "Kind": 77,
      "Time": "2018-09-05T10:56:37+02:00"
    },
    {
      "Kind": 78,
      "Time": "2018-09-05T10:56:45+02:00"
    },
    {
      "Kind": 71,
      "Time": "2018-09-05T10:59:11+02:00",
      "Text": "TAN"
    },
    {
      "Kind": 75,
      "Time": "2018-09-05T11:06:21+02:00"
    },
    {
      "Kind": 6,
      "Time": "2018-09-05T11:06:21+02:00"
    },
    {
      "Kind": 23,
      "Time": "2018-09-05T11:06:21+02:00"
    },
    {
      "Kind": 71,
      "Time": "2018-09-05T10:59:11+02:00"
    },
    {
      "Kind": 95,
      "Time": "2018-09-05T11:09:51+02:00"
    }
  ]
}
```

6.4 getIdentData

The usual next step after an OnFinal webhook and the getStatus request is the getIdentData request with the subsequent logic:

6.4.1 getIdentData Requirements

Final status kinds and data availability for identification order without QES

Status Code	Status Kind	Identdata Availability
6	Positive identification	Yes
17	Negative identification	Depends
22	Manually terminated	No
89	Fraud Suspicion	Depends
99	TimeOut	No

Final status kinds and data availability for identification order with QES

Status Code	Status Kind	Identdata Availability
84	Document signed	Yes
17	Negative identification	Depends
22	Manually terminated	No
89	Fraud Suspicion	Depends
99	TimeOut	Depends
100	TimeOut after identification	Yes (6 without QES)

6.4.2 getIdentData URL

[GET] <https://customer.identity.tm/api/2.09/getIdentData/OrderID/IncludeInitialData>

[GET] <https://customer.identity.tm/api/2.09/getIdentData/OrderID/IncludeIdentifyMethod>

[GET] <https://customer.identity.tm/api/2.09/getIdentData/OrderID/Signed>

[GET] <https://customer.identity.tm/api/2.09/getIdentData/OrderID>

6.4.3 getIdentData variants

Suffix	Description
<i>none</i>	Normal variant, response as described below
IncludeInitialData	Include initial data as given to the putOrder request
IncludeIdentifyMethod	Include an additional element in the response "IdentifyMethod"
Signed	Return signed and encrypted version of the identification data for later re-use

6.4.4 getIdentData Response Data

Element	Superior element	Data type	Length	Mandatory	Description
OrderID	[ROOT]	String	20	Yes	Requested OrderID
Ref	[ROOT]	String	100	Yes	Corresponding Ref
IdentifyMethod	[ROOT]	enum		Depends	Method in which the identification occurred: Courier, Shop, Video, eID, Giro
Identified	[ROOT]	DateTime		Yes	Time of identification
IdentData	IdentData	Object		Yes	
Type	IdentData	Int		No	1 = checked Data 2 = initial Data
Firstname	IdentData	String	50	Yes	
Surname	IdentData	String	50	Yes	
Street	IdentData	String	100	Yes	
ZipCode	IdentData	String	5	Yes	
City	IdentData	String	100	Yes	
Country	IdentData	String		Depends	
Birthdate	IdentData	Date		Yes	
Birthplace	IdentData	String	100	Yes	
Nationality	IdentData	String	50	Depends	
Title	IdentData	String	50	Depends	
Artistname	IdentData	String	50	Depends	
Gender	IdentData	String		Depends	female/male
Mobile	IdentData	String	50	Depends	If verified with TAN
Email	IdentData	String	100	Depends	If verified with TAN

IDCard	IdentData	Object			
CardNo	IDCard	String	50	Depends	
PlaceOfIssue	IDCard	String	100	Depends	
ValidUntil	IDCard	Date		Depends	
Type	IDCard	String		Depends	IDCard/Passport or LegitimationDocument provided in putOrder
DriverLicence	IdentData	Object		Depends	If enabled in putOrder
LicenceNo	DriverLicence	String	50		If enabled in putOrder
Classes	DriverLicence	String	50		
DateOfIssue	DriverLicence	Date			
DateOfIssue	DriverLicence	Date			
Additional	IdentData			No	If set in putOrder
<detail>	Additional	String	150	No	If set in putOrder
Images	IdentData	Array		Depends	If enabled for this request
Image	Images	String			Base64 encoded Data
Filename	Image	String	100		Filename for the data

6.4.5 getIdentData HTTP Codes

200 plus JSON document

401 if customer authentication fails, please check username and password

400 other error: JSON document with error in body

404 if not found

6.4.6 getIdentData Example Request

```
[GET] https://customer.identity.tm/api/2.0/getIdentData/99921269855041/IncludeInitialData
```

6.4.7 getIdentData Example Response

```
{
  "Identitfied": "2018-09-05T11:06:21+02:00",
  "IdentData": [
    {
      "Type": 1,
      "Firstname": "Erika",
      "Surname": "Mustermann",
      "Street": "Heidestr. 17",
      "ZipCode": "43000",
      "City": "Köln",
      "Birthdate": "1964-08-12",
      "Birthplace": "Berlin",
      "Nationality": "DE",
      "Gender": "female",
      "Mobile": "+4915114084205",
      "IDCard": {
        "CardNo": "T22000129",
        "PlaceOfIssue": "Stadt Köln",
        "ValidUntil": "2020-10-31",
        "Type": "IDCard"
      }
    },
    {
      "Type": 2,
      "Firstname": "Erika",
      "Surname": "Mustermann",
      "Street": "Heidestr. 17",
      "ZipCode": "43000",
      "City": "Köln",
      "Birthdate": "1964-08-12",
      "Birthplace": "Berlin",
      "Nationality": "DE",
      "IDCard": {
        "CardNo": null,
        "PlaceOfIssue": null,
        "ValidUntil": "1970-01-01"
      }
    }
  ]
}
```

6.5 getIdentDataPDF

For this result file, the same getIdentData logic applies as in chapter 6.4. The content is basically the same except this is pdf format and all images/screenshots are included by default.

6.6 getIdentDataPDF Requirement

Same as getIdentData, additionally this endpoint must be enabled by identity.tm.

6.6.1 getIdentDataPDF URL

[GET] <https://customer.identity.tm/api/2.09/getIdentDataPDF/OrderID>

6.6.2 getIdentDataPDF HTTP Codes

200 plus PDF Stream

401 if customer authentication fails, please check username and password

400 other error: JSON document with error in body

404 if not found

6.6.3 getIdentDataPDF Example Request

[GET] <https://customer.identity.tm/api/2.0/getIdentDataPDF/99921269855041>

6.6.4 getIdentDataPDF Example Response

Resultat IDENTprüfung

Auftrag: 99921267754340

Referenz: Shop GWG

Angabe	Geliefert	Geprüft
Nachname	Mustermann	Mustermann
Vorname	Erika	Erika
Strasse	Heidestr. 17	Heidestr. 17
PLZ	51147	51147
Ort	Köln	Köln
Geburtsdatum	12.08.1964	12.08.1964
Geburtsort	Berlin	Berlin
Nationalität		DE
Künstlername		
Titel		
Ausweisnummer		T22000129
AusstellungsLand		
AusstellungsOrt		Stadt Köln
Gültig bis		31.10.2020

Lichtbild und Merkmale gemäß Ausweisdokument mit der zu identifizierenden Person verglichen und Identität festgestellt.

Resultat
Identifizierer
Prüfer

positiv am 03.09.2018 08:52

1872

06.09.2018 16:10 (1/3)

6.7 getESignPDF

Endpoint for fetching processed/signed eSign PDF documents

6.8 getESignPDF Requirement

If eSignPDF element was set in putOrder and final status is 84 - kinds and data availability for identification order with QES:

Status Code	Status Kind	ESignPDF Availability
84	Document signed	Yes
17	Negative identification	No
22	Manually terminated	No
89	Fraud Suspicion	No
99	TimeOut	No
100	TimeOut after identification	No

6.8.1 getESignPDF URL

[GET] <https://customer.identity.tm/api/2.09/getESignPDF/OrderID>

6.8.2 getESignPDF Response Data

Element	Superior element	Data type	Length	Mandatory	Description
PDFs	[ROOT]	Array			
Base64Data	PDFs	String			

6.8.3 getESignPDF HTTP Codes

200 plus PDF Stream if only one Document, otherwise a JSON document

401 if customer authentication fails, please check username and password

400 other error: JSON document with error in body

404 if not found

6.9 getESignHash

Endpoint for fetching processed/signed eSign Hashes

6.9.1 getESignHash Requirement

If eSignHash element was set in putOrder and final status is 84 - kinds and data availability for identification order with QES:

Status Code	Status Kind	ESignHash Availability
84	Document signed	Yes
17	Negative identification	No
22	Manually terminated	No
89	Fraud Suspicion	No
99	TimeOut	No
100	TimeOut after identification	No

6.9.2 getESignHash URL

[GET] <https://customer.identity.tm/api/2.09/getESignHash/OrderID>

6.9.3 getESignHash Response Data

Element	Superior element	Data type	Length	Mandatory	Description
Hashes	[ROOT]	Array			
<Name>	Hashes	String			The Name given while placing the Order

6.9.4 getIdentData HTTP Codes

200 plus JSON document

401 if customer authentication fails, please check username and password

400 other error: JSON document with error in body

404 if not found

6.10 getVideoFileBinary

This file is provided, when identification was conducted in video with a recording of the session from the agent's point of view.

6.10.1 getVideoFileBinary URL Requirement

This file is provided, when final status from video is 6 or 89 and the recording requirement was set either in the Add element of the putOrder request or globally configured in the account for legal (e.g. German AML) compliance by identity.

6.10.2 getVideoFileBinary URL

[GET] <https://customer.identity.tm/api/2.09/getVideoFileBinary/OrderID>

6.10.3 getVideoFileBinary HTTP Codes

200 plus video/mp4

401 if customer authentication fails, please check username and password

400 other error: JSON document with error in body

404 if not found

6.11 getVoiceFiles

In some countries it is sufficient, to have an audio recording instead of a video file. In these cases the configuration will be made by identity for the given account. An audio recording cannot be requested with the putOrder request.

6.11.1 getVoiceFiles URL

[GET] <https://customer.identity.tm/api/2.09/getVoiceFiles/OrderID>

6.11.2 getVoiceFiles HTTP Codes

200 plus JSON document

401 if customer authentication fails, please check username and password

400 other error: JSON document with error in body

404 if not found

6.11.3 getVoiceFiles Example Request

```
[GET] https://customer.identity.tm/api/2.0/getVoiceFiles/99920006C38842
```

6.11.4 getVoiceFiles Example Response

```
{"Voicefiles":[{"Filename":"sound_4348.mp3","Base64Data":"\\uQBAAAAA....."}]}
```

6.12 delIdentdata

By default, identity will delete all result files 7 days after fulfilment. Nevertheless, it would be best, to delete the result files after they have been collected completely and successfully.

Some confusion exists on when to use delIdentData and when to use cancelOrder. As a rule of thumb, use cancelOrder before the OnFinal webhook has been triggered and delIdentData after that point.

6.12.1 delIdentdata URL

[DELETE] <https://customer.identity.tm/api/2.09/delIdentdata/OrderID>

6.12.2 delIdentdata HTTP Codes

202 Accepted

401 if customer authentication fails, please check username and password

400 other error: JSON document with error in body

404 if not found, OrderID invalid or no ident data available

406 if the Order is in state that's not valid to be cancelled

7 How to handle incomplete Orders

7.1 Setting up a TimeOut

Please contact itsupport@identity.tm in order to set a general TimeOut for incomplete orders.

7.2 getAllStatus

Endpoint to get all Scans/Status to all orders. If the Element period isn't given, all Status that are not responded by that method before (without period) will be responded. If you want POST json Data, use content-type json. Otherwise not.

7.2.1 getAllStatus URL

[POST] <https://customer.identity.tm/api/2.09/getAllStatus/ExtendedList>

[POST] <https://customer.identity.tm/api/2.09/getAllStatus>

7.2.2 getAllStatus POST Data (optional)

Element	Superior element	Data type	Length	Mandatory	Description
Period	[ROOT]			No	If Element exists, Status will not be marked as responded
Begin	Period	DateTime			Status from
End	Period	DateTime			Status until
Paging	Period			No	For Paging the result if Result >= 1000 Elements
Elements	Paging	Int			<= 1000
Page	Paging	Int			

7.2.3 getAllStatus Response Data

Maximum 1000 Status elements get returned at once

Element	Superior element	Data type	Length	Mandatory	Description
Order	[ROOT]				
OrderID	Order	String	20	Yes	The OrderID
Ref	Order	String	100	Yes	Your internal number to the order
Delivered	Order	DateTime		Depends	DateTime if Order is delivered/executed ISO 8601
Status	Order	Array			
Kind	Status	Int		Yes	See Description of <u>kind of status</u> (at the end of this Document)
Time	Status	DateTime		Yes	ISO 8601
Text	Status	String	255	Depends	Hints to a status
TVSlot	Order	Object		Depends	Current TV – just in Status 19
SlotBegin	TVSlot	DateTime		Yes	Begin of Slot ISO 8601
SlotEnd	TVSlot	DateTime		Yes	End of Slot ISO 8601

7.2.4 getAllStatus HTTP Codes

HTTP Code 200 + JSON

HTTP Code 304 if no new Status

HTTP Code 401 if customer validation fails

HTTP Code 400 other error (JSON-Document – Error – in Body)

7.2.5 getAllStatus Request Example

```
{
  "ExtendedList": true,
  "Period": {
    "Begin": "2016-01-01T00:00:00+01:00",
    "End": "2016-01-15T00:00:00+01:00",
    "Paging": {
      "Elements": 200,
      "Page": 2
    }
  }
}
```

7.2.6 getAllStatus Response Example

```
[
  {
    "OrderID": "9992120505XXXX",
    "Ref": "TestOrder"
  },
  {
    "OrderID": "9992000BE9XXXX",
    "Ref": "Test",
    "Executed": "2018-05-28T16:47:00+02:00"
  },
  {
    "OrderID": "9992121170XXXX",
    "Ref": "TestOrder",
    "Executed": "2018-05-30T16:05:52+02:00"
  },
  {
    "OrderID": "9992121171XXXX",
    "Ref": "TestOrder",
    "Executed": "2018-05-30T16:08:52+02:00"
  }
]
```

7.3 cancelOrder

If an open order shall not be processed anymore, you can cancel it with this endpoint. The order data will not be deleted, but the user will not be able to proceed with his identification or signature.

7.3.1 cancelOrder URL

[POST] <https://customer.identity.tm/api/2.09/cancelOrder/OrderID>

7.3.2 cancelOrder Example URL

<https://customer.identity.tm/api/2.09/cancelOrder/99920006C38842>

7.3.3 cancelOrder Response Data

HTTP Code **202** Accepted

HTTP Code 401 if customer validation fails

HTTP Code 404 if the order cannot be found

HTTP Code 406 if the order is in state that is not valid to be cancelled

HTTP Code 400 other error /JSON-Document – Error – in Body)

8 Re-using identifications to sign documents

If your account is eSign-enabled and is using Namirial, it is possible to store identification data and re-use it later to avoid a going through a whole new identification process.

To be able to do so, a two-step approach is used:

1. Retrieve and store a signed version of the identification data
2. Provide this identification data when placing a new order for the same user

To retrieve the signed version of the identification data you need to call the “Signed”-variant of `getIdentData` (`getIdentData/OrderID/Signed`) as described in chapter 6.4 and store the result body.

When you now want to sign a document and re-use the identification, you provide the stored data in the element `IdentDataSigned` inside the body of the `putOrder` call.

Before re-using the given data, it will be checked, if

- It has not been tampered and is signed correctly,
- The given data is sufficient to be used for signing purposes,
- The identity document used for the identification is still valid.

In case the data is considered valid for signing purposes, the user will immediately be redirected to the signing process without a further identification, otherwise the user will be re-identified.

9 Monitoring

9.1 serverStatus

Method to check the Service. Will return HTTP Code 200 without any Data.

URL = <https://customer.identity.tm/api/2.09/serverStatus>

10 IP-Ranges currently used

Range	Network name
212.211.189.136 - 212.211.189.143	ID8_NET
212.211.206.240 - 212.211.206.247	identity-NET
212.211.181.64 - 212.211.181.71	idvos_NET
212.211.191.96 - 212.211.191.103	ID8_NET

11 Technical requirements for Video Ident

- Browser requirement
 - Desktop: Chrome, Opera, Firefox, Safari
 - Mobile: Android, Chrome - latest versions (official), alternatively mobile apps for Android and iOS can be used
- Bandwidth: Minimum 0,5 MB/s up/down
- Camera: Minimum resolution 640 x 480 px
- Network requirements:
 - Minimum: The minimum Requirement is that TCP port 443 is open. Some firewall/proxy rules only allow for SSL traffic over port 443. You will need to make sure that non-web traffic can also pass over this port. TLS1.2
 - Better Experience: In addition to the minimum requirements being met, we also recommend that UDP port 3478 is open. TLS1.2
 - Best Experience: For the best possible experience, we recommend that UDP ports 1025 - 65535 be open. TLS1.2
- WebRTC: Outbound TCP, non-SSL web traffic on port 443 and the following domains must be accessible:
 - *.tokbox.com (static IP blocks also available)
 - static.opentok.com
 - enterprise.opentok.com
 - api.opentok.com
 - anvil.opentok.com
- WebSocket: In some situations WebSocket connections are blocked over port 80. In this case a secure SSL connection using WSS over port 443 should successfully connect. The destinations and ports used by Pusher clients are as follows:
 - ws://ws.pusherapp.com on port 80
 - wss://ws.pusherapp.com on port 443