Advania Janus

Utilities v2 api

Advania

CONTENTS

1 INTRODUCTION

Janus API is a utilities service for a variety of functions needed by applications i.e. certificate authentication, national registry, document alterations, signature validation and sending SMS messages. This document describes some of these functions.

2 SERVICE URLS

The service is available in testing/development and production environment. The URLs are

- Testing:
  - https://prufa.signet.is/utilitiesservice/v2
- Production:
  - https://traust.vottun.is/utilitiesservice/v2

The service has a API help and swagger description pages which help with integration.

3 AUTHENTICATION AND MONITORING

The api uses basic authentication where the user/password combination is supplied by the Signet department.

Every controller has GET ping function in the form of api/<controller>/ping?message=<message> where <controller> is the name of the controller and <message> is a message you supply. If the service is running and the authentication is successful the function will return the following string with the message you supplied.

Success: Message: <message>

4 CONTROLLERS

The API has multiple controllers for various functions. This chapter describes these controllers and functions.

4.1 EVROTRUST

The evrotrust contoller is used for requesting authentication and signature via the Evrotrust app (https://www.evrotrust.com/). The path for the controller is <url>/api/evrotrust/<function>.

It is recommended to use users full mobile phone number for requests (i.e. +3544409000) and start with the EtCheckUser functions to be sure a user is registered with Evrotrust.

4.1.1 ETCHECKUSER

The EtCheckUser function accepts a EvrotrustRequest and returns a boolean true or false depending if the user is registered with Evrotrust and ready to authenticate/sign or not.

Example:

Request:

POST: /EtCheckUser


```
{
  "identificationNumber": null,
  "country": null,
  "email": null,
  "phone": "+3544409000"
}
```


Response: „true"

4.1.2 ETCHECKUSEREXTENDED

The EtCheckUserExtended function is also used to determine if a user is registered with Evrotrust and ready to authenticate and sign. The response however is more detailed of the form of EvrotrustUserResponse which further describes information about Evrotrust user identification status: is registered, is identified, has confirmed phone, has confirmed email and if identification is confirmed by supervisor

Example:

POST: /EtCheckUserExtended


```
{
  "identificationNumber": null,
  "country": null,
  "email": null,
  "phone": "+3544409000"
}
```


Response:


```
{
 "Registered": true,
 "Identified": true,
 "Rejected": false,
```

"Supervised": true,

"ReadyToSign": true,

"ConfirmedPhone": true,

"ConfirmedEmail": true

}

Description of response:

"Registered": true

The user has started Evrotrust registration process and has scanned an ID card. To sign documents the user should proceed to successful face recognition and validation of a phone number.

"Identified": true

The user has has passed through Evrotrust registration and has been successfully identified. To sign documents he/she should also validate a phone number.

"Rejected ": true

The user has been rejected by the Operator/Supervisor. The reason may be invalid, expired, broken ID document, fake ID, different person on the video and the ID.

"Supervised": true

The user completed all required steps for profile creation. Live supervisory review over the profile has passed successfully. The user can sign documents.

"ReadyToSign": true

The user completed the minimum required steps for profile creation. For Bulgarian citizens it is activated automatically after a successful face recognition (before "Supervised": true). The user can sign documents.

"ConfirmedPhone": true

The user has completed Evrotrust registration, identified successfully and have validated phone number. The user can sign documents.

"ConfirmedEmail": true

The user completed Evrotrust registration, was successfully identified, verified phone number and e-mail address. The Email address is not obligatory in Evrotrust and users can operate with or without it. The user can sign documents.

4.1.3 ETCHECKCERTIFICATE

This function is used to check if a certificate returned from authentication belongs to the same user (used when user gets a new certificate and doesn‘t contain users personal number but the Evrotrust P-number which changes for every new certificate). It will return true if the certificate belgongs to the same user.

Example:

POST: /EtCheckCertificate

```
{
  "identificationNumber": null,
  "country": null,
  "email": null,
  "phone": "+3544409000",
  "serial": "1234",
  "cert": "<base64 string>",
}
```

Response: „true"

4.1.3.1 ETGETLATESCERTIFICATE

The EtGetLatesCertificate function is used to get a users latest certificate depending on the request paramaters. This is primarily used for signatures and returns a EvrotrustCertificate response.

Example:

Post /EtGetLatestCertificate

```
{
  "identificationNumber": null,
  "country": null,
  "email": null,
  "phone": "+3544409000",
  "coverage": 0,
```

```
  "qualified": true,
  "pid": true
}
```

Response:

```
{
  "type": 1,
  "certificate": "<base64 encoded certificate>...",
  "coverage": 20000,
  "dateValidTo": 1765455488,
  "serialNumber": "0AAC52B6525BAA6DE3655BEEC3EC7EF9B77C3401"
}
```

coverage "integer" 0, 500, 2000, 20000, 100000, 250000;

pid "boolean" whether the personal identifier is included in the certificate or not.

type "integer" valid values 1 - E-Sign Qualified; 2 - E-Sign Advanced;

4.1.4 ETAUTHENTICATE

The function EtAuthenticate is used to authenticate a user via async method where the returned transid is used to poll for status (/EtGetStatus) and then download of the authentication (/EtDownload). The method accepts a EvrutrustAuthRequest which is as follows

"identificationNumber": "string", the users id number,

"country": "string", the users country

"email": "string", the users email

"phone": "string", the users mobile

"desc": "string", a description visible in the users app

"requireSigning": true, require the user to sign the authentication request

Example:

POST /EtAuthenticate

```
{
```

```
    "identificationNumber": null,
    "country": null,
    "email": null.
    "phone": "+3544409000",
    "desc": "Test description",
    "requireSigning": true
}
```

Response:

"433961318796"

4.1.5 ETGETSTATUS

The EtGetStatus function is used to poll for the status of an authentication request and returns an EvrotrustStatus object describing the status of transaction.

Example:

GET /EtGetStatus?transId=433961318796

Response

```
{
    "Ready": false,
    "Status": 1,
    "Reason": null
}
```

Here the authentication is waiting on the user to accept

GET /EtGetStatus?transId=433961318796

Response

```
{
    "Ready": false,
    "Status": 2,
    "Reason": null
}
```

Here the user has accepted/signed and Evrotrust is finishing creating the authentication XML.

GET /EtGetStatus?transId=433961318796

Response:

```
{
  "Ready": true,
  "Status": 2,
  "Reason": null
}
```

Here the authenticaton is ready and a call to EtDownload with the transid can be made to download the authentication.

The possible Status values are:

1 – Pending – awaiting signature from the user

2 – Signed – the user signed the document

3 – Rejected – the user rejected signing the document

4 – Expired – the time expired and the document is no longer available for signing

5 – Failed – a communication error occurred and signature is impossible. Resend the file for signature

6 - Withdrawn - the sender withdrew the documents and it is no longer available for signature

7 - Undeliverable - status 99 goes to a final status 7 if the user stays unregistered within the document expiration time

99 – On hold – The document is not yet available for signature. The most common cases are: If the document has been sent to several users, the previous signature is still undergoing; The user has not yet downloaded Evrotrust app and has not yet completed the registration proceess.

If the user rejected the request and entered a reason it will be shown in the Reason field.

Example:

```
{
  "Ready": false,
  "Status": 3,
  "Reason": "Hafnað"
}
```

4.1.6 ETDOWNLOAD

EtDownload is used to download the data after successful authentication or confirmation request.

Example

GET /EtGetStatus?transId=433961318796

Response:

"PD94bWwgdmVyc2lvb….”

Base64 endoded XAdES LTV file with the users certificate and chain.

4.1.7 ETSIGN

The EtSign function is used to request a signature from a user. This is not available to 3rd party service providers.

4.1.8 ETCONFIRMDATA

The EtConfirmData function is used to make the user sign a document showing his initial Evrotrust authentication data. This is not available to 3rd party service providers.

4.1.9 ETAUTHENTICATESYNC

The EtAuthenticateSync function is similar to EtAuthenticate but is a sync method so it will return the base64 encoded authentication data if successful. We do not recommend using this method.

4.2 MISC

The misc controller has various helper functions as well as getting users digital certificates from Auðkenni. The path for the controller is <url>/api/misc/<function>

Further description soon...

4.3 POWERS

The powers controller is used for searching an individuals powers and/or certifications. The path for the controller is <url>/api/powers/<function>

4.3.1 GETMEDICAL

For requesting medical license of individual a GET request to /getmedical?ssn=<SSN> is needed, where <SSN> is the individuals national registry number. The functions return a MedicalPowers object with furter information if individual is found.

4.4 REGISTRY

The registry controller is used for Icelandic national registry lookup. The path for the controller is <url>/api/registry/<function>

Further description soon...

4.5 SHIP

The ship controller is used for lookup of Icelandic ships. The path for the controller is <url>/api/ship/<function>

4.5.1 GETSHIP

For requesting info on a ship by registry number a GET request to /getship?number=<number> is needed where <number> is the ships registration number. The functions returns a ShipInfo object with further informatin if ship is found in registry.

4.6 SMARTID

The smartid controller is used for authentication with the Auðkenni app (https://app.audkenni.is). The path for the controller is <url>/api/smartid/<function>.

The Auðkenni app requires a SHA512 hash to be signed for authentication and will display a 4 digit verification code to the user which is calculated from the supplied hash. There are some helper functions supplied to make this integration easier which will be described below.

4.6.1 GETVCCODE

To get a vc code for a supplied hash a GET request to /getvccode?hash=<hash> is needed where <hash> is the SHA512 hash to be used. The function returns a string with a 4 digid verification code if succesful.

4.6.2 GETRANDOMHASH

To get a random SHA512 hash and a corresponding VCCode a GET request to /getrandomhash is needed. The function will return a HashResponse object with a random hash and corresponding vccode.

Example:


```
{
  "Hash":
"mMZKVKlAYOrEbqqrt+8cxAywmo6PqwICmsHW3qL+P0W/EkqU/wVbZT5X4M4WsA8U9Jzk
+uQFTwU7n8vemIQvJw==",
  "VCcode": "4432"
}
```


4.6.3 AUTHENTICATE

To authenticate a user with a sync call a POST request with a SmartIDAuthRequest object with a supplied hash to /authenticate is required. If the authentication was succesful a SmartIDResponse

object is returned with the authentication data but if the request was unsuccessful a ResultStatus object is returned with further information.

4.6.4 AUTHENTICATEGETCODE

To authenticate a user with an async call a POST request with a SmartIDAuthRequest object to /authenticategetcode is needed. Here the hash is not required and if not supplied the api will generate a random hash which will be included in the SmartIDAuthresponseResponse as well as the vccode and sessionid.

Example:

```
{
  "SessionID": "XwFaKWxgnsq4X5YItvDMWjC4umA",
  "Hash":
"zV8tp+eB3nmulfED8OpcR7sS53ckY5J77mX+7R6MGS6Ftxz+mej8V3/mRsq/E8pXYki+
if2Gwg6OCjHGOHtX9g==",
  "VCcode": "2709"
}
```

4.6.5 GETSESSIONSTATUS

To get the status of authentication from AuthenticateGetCode a GET request to /getsessionstatus/<id> is required where <id> is the sessionid obtained from authenticategetcode. If the user has authenticated a CIBAResponse is returned with information about the authentication and user. If the user has not authenticated yet or cancelled a HTTP 400 response with ResultStatus is returned with further information and error codes. See appendix for codes.

4.6.6 SELECTCERTIFICATESYNC

To get a user certifiate (for signature) in a sycn call a POST request to SelectCertificateSync?ssn=<ssn> request is required where <ssn> is the users national registry number. If the user has only one active certificate no interaction of the user is needed, else the user has to confirm the action on the appropriate device. The function returns a CIBAResponse if successful.

4.6.7 SELECTCERTIFICATES

To get a user certificate (for signature) a POST request to SelectCertificateSync?ssn=<ssn> request is required where <ssn> is the users national registry number. If the user has only one active certificate no interaction of the user is needed, else the user has to confirm the action on the appropriate device. The function returns a session ID to use in GetSessionStatus for results.

4.7 TOKEN

The token controller is used for token actions such as generating SAML or JWT tokens. The path for the controller is <url>/api/token/<function>

Further description soon...

4.8 VALIMO

The Valimo is used for authentication with the Auðkenni SIM certificates. The path for the controller is <url>/api/valimo/<function>.

4.8.1 PHONELOGIN

To authenticate a user in a sync call a GET request to PhoneLogin?phonenumber=<number>&authText=<text> is required where <number> is the users mobile number and <text> is the message to display on the users mobile. If successful the function returns the users certificate in a base64 string.

4.8.2 PHONELOGIN2

To authenticate a user in a sync call a GET request to PhoneLogin2?phonenumber=<number>&authText=<text> is required where <number> is the users mobile number and <text> is the message to display on the users mobile. If successful the function returns UtCertInfo object with information about the users certificate and the certificate.

4.8.3 PHONELOGINXML

To authenticate a user in a sync call a GET request to PhoneLoginXml?phonenumber=<number>&authText=<text> is required where <number> is the users mobile number and <text> is the message to display on the users mobile. If successful the function returns XmlCertInfo object with a signed XML document and information about the users certificate and the certificate.

4.8.4 GETMOBILECERTIFICATE

To get a users mobile signature certificate a GET request to getmobilecertificate?ssn=<ssn>&mobile=<mobile> is required where <ssn> is the users national registry number and <mobile> his mobile (with 354 prefix). If succesful the function returns the information in ISCertInfo object.

4.8.5 PHONELOGINPOLL

To authenticate a user in an async call (polling) a GET request to PhoneLoginPoll?phonenumber=<number>&authText=<text>&xml=<xml> is required where <number> is the users mobile number, <text> is the message to display on the users mobile and <xml> is true or false indicating if response should be a signed xml. If successful the function returns the transaction id to be used in GetValimoStatus.

4.8.6 GETVALIMOSTATUS

To fetch the authentication status of PhoneLoginPoll above a GET request to GetValimoStatus/<id> is required where <id> is the transaction id. If successful a ValimoAuthResponse is returned with information about user, certificate and authentication is returned. If not successful a ResultStatus object is returned with further information on error or status.

5 APPENDIX – AUÐKENNI TEST USERS AND ERROR CODES

5.1 VALIMO/SIM CERTIFICATES

5.1.1 TEST USERS

Name	SSN	Mobile
Test Notandi	1234567890	+3541111111
Test Notandi 2	1234567899	+3541111112
Test Notandi 3	1234567891	+3541111113
Test Notandi 4	1234567892	+3541111114
Test Notandi 5	1234567893	+3541111115
Test Notandi Cancel	0987654321	+3542222222
Test Notandi Problem	0202021234	+3544444444
Test Notandi Timeout	0101011234	+3543333333

5.1.2 STATUS CODES

Code	Description
100	The request was accepted by Valimo Signature Server.
500	A Mobile Signature has been successfully constructed and is available.
501	A Mobile Signature has been successfully constructed. However, the signer's certificate is revoked.
502	A Mobile Signature has been successfully constructed and the signature is valid.
503	Invalid Signature
504	The Transaction is still being processed.

5.1.3 ERROR CODES

Code	Description
101	Wrong parameter. Error among the request arguments
102	Missing parameter. An argument is missing from the request.
103	Wrong data length. A field in the request contains too long data.
104	Unauthorized access. The AP is unknown or the password is wrong.
105	Unknown client. The end user targeted by the AP is unknown to Valimo Signature Server.
107	Inappropriate data. Valimo Signature Server cannot handle the given data.
108	Incompatible interface. The minor version or the major version parameter is inappropriate or the request is not supported.
109	Unsupported profile. The AP has specified a mobile signature profile that is not supported.
208	Expired transaction. Transaction expiry date has been reached, or a timeout has

	elapsed.
209	OTA error. Valimo Signature Server has not succeeded to contact the end user's mobile equipment.
401	User Cancel. The end-user has cancelled the signing or already in an other transaction
402	PIN blocked. The PIN for the key to be used has been blocked
403	Card Blocked. The SIM (or signing PUK) has been blocked
404	No Key Found. Signing was requested for a key that does not exist
422	No certificate. No certificate has been found for this MSISDN.
425	Error certificate. Error in certificate validation.
900	Internal error. An internal error has occurred in Valimo Signature Server.

5.2 AUÐKENNI ERROR CODES

5.2.1 MOBILE ERROR CODES

Code	IS description	EN description
mssp_100		Got accepted state when checking the status. This should not happen
mssp_101	Beiðni ekki rétt	Wrong parameter
mssp_102	Vantar breytu	Missing parameter
mssp_103	Svæði í beiðni of stórt	Wrong data length
mssp_104	Þjónustuveitandi ekki þekktur	AP unknown
mssp_105	Notandi finnst ekki.	User not found
mssp_107	Ekki hægt að vinna úr beiðni	Unable to handle given data
mssp_108	Misræmi í útgáfum	Incompatible interface version.
mssp_109	Óþekktur prófíll	Unsupported profile
mssp_208	Rann út á tíma	Timeout - Ekkert svar frá síma
mssp_209	Rann út á tíma	Timeout - Næst ekki í síma
mssp_401	Notandi hætti við	User cancel
mssp_402	PIN læst	PIN blocked
mssp_403	Kort læst	Card blocked
mssp_404	Skilríki finnast ekki	No key found
mssp_422	Skilríki finnast ekki	No certificate found
mssp_425	Villa við sannreyningu	Error in certificate validation
mssp_501	Skilríki er afturkallað	Certificate is revoked
mssp_503	Villa við sannreyningu	Error in signature verfication
mssp_504	Beiðni í vinnslu	Request in progress
mssp_900	Villa í innri kerfum	Error

5.2.2 SMART ID ERROR CODES

Code	IS description	EN description
SmartID_467	Notandi hætti við	User cancel
SmartID_468	Rann út á tíma	Timeout

SmartID_469	Villa í samskiptum	Communication error
SmartID_470	Rangur staðfestingarkóði valinn	Wrong verification code selected
SmartID_471	Notandi finnst ekki	User not found
SmartID_472	Villa - athugið Auðkennis APP	Error - Check Audkennis APP
SmartID_480	Misræmi í útgáfum	Incompatible interface version.
SmartID_580	Viðhaldsvinna í gangi, reynið aftur síðar	System is under maintenance, retry later
SmartID_900	Villa í innri kerfum	Error
1	Notandi hætti við	User cancel
404	Notandi finnst ekki	User not found
408	Rann út á tíma	Timeout
501		Invalid certificate