Signant.no
Web service
Authentication
Functional specification

19. juni. 2017

Version 1.3

Author: Thorbergur Gislason – tg@maestro.no

Contents

Version log 2
1. About Signant.no 2
2. Introduction 3
2.1 Purpose of this document. 3
2.2 Target audience. 3
2.3 Scope of the document 3
2.4 Definitions, Acronyms and Abbreviations 3
3. Data Flow 3
3.1 Dataflow steps 4
4. Web service functions 4
4.1 InitAuth 4
4.2 GetAuthStatus 5
5 AuthOptions 6
6 States and status codes 6
6.1 Status codes 6
6.2 Session state 6
7 Sample requests 7
7.1 InitAuth sample request 7
7.2 GetAuthStatus sample request 7

Version log

Version	Date	Changes
1.0	2015.04.10	Initial document
1.1	2017.02.15	Small changes and adjustments
1.2	2017.04.28	Added AuthType NO_BankID_Mobile
1.3	2017.06.19	UserFirstName and UserLastName added to GetAuthStatus result

1. About Signant.no

Signant.no is created and owned by Maestro Soft AS. It provides services for electronic signing and exchange of documents in addition to providing a secure authentication service with multiple electronic identities.

2. Introduction

2.1 Purpose of this document.

Describe the technical solution for creating and validating authentication requests using the signant.no Authentication WS API.

2.2 Target audience.

The target audience for this document are IT professionals and developers that want to implement Signant authentication.

2.3 Scope of the document

This document is meant to describe available functions and dataflow between customer and signant.no system.

2.4 Definitions, Acronyms and Abbreviations

The following table presents the most significant terms used in this document:

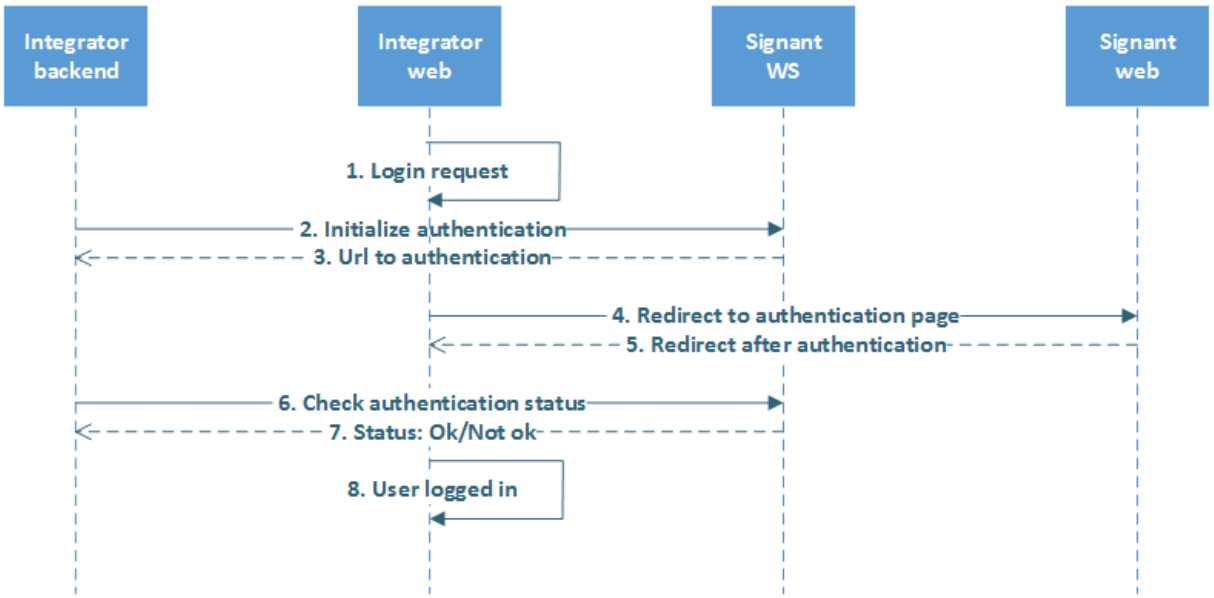
Term	Definition
Signant	Service provider
Integrator	Company using the WS
User	Person being authenticated
Integrator web	Web page being accessed
Integrator backend	Part of the system that talks to the Signant WS
Signant WS	Web service to initialize and validate the authentication session
Signant Web	Web page where the user inputs his credentials

3. Data Flow

The system works in two layers, the WS layer for initializing and validating the authentication, and web layer where the user interacts with the service.

The integrator communicates with the “Signant authentication service” by a secure WS connection. Integrator must take special care not to make critical data from the WS communication (DistributorID, AccessCode, TrackingID) possible to see, lookup or detect by network sniffing tools on the client machine.

The image below illustrates the most common use of Signant Authentication.



3.1 Dataflow steps

1) User tries to log on to Integrator web.
2) Integrator initializes new Authentication.
3) URL to authentication is returned.
4) User is redirected to authentication page.
5) After successful authentication the user is redirected back to Integrator web.
6) The redirect back (step 5.) should trigger a call to the “authentication status check” function call from the Integrator system.
7) Signant WS returns authentication status.
8) If authentication was successful the user is logged in to the Integrator web.

4. Web service functions

This web service exposes the following functions:

Name	Description
InitAuth	For creating a new authentication session.
GetAuthStatus	Returns the status for the authentication session.

4.1 InitAuth

Used to create a new authentication session. See 7.1 for sample request.

InitAuth parameters:

Name	Type	Description	Required
DistributorID	String	For authentication (Provided by Signant)	Yes
AccessCode	String	For authentication (Provided by Signant)	Yes
AuthOptions	AuthOptions	Options for the Auth session	Yes

InitAuth : AuthOptions

Name	Type	Description	Required
AuthTypes	`AuthType[]`	Array of authentication types to be made available to the user 1)	Yes
NextUrl	String	Redirect Url after successful authentication 2)	Yes
CancelUrl	String	Redirect Url after cancelled or failed authentication	Yes
ReturnSSN	Bool	Return SSN after authentication 3)	Yes

1) See AuthOptions for a list of available authentication types
2) requestid will be added to the querystring before redirect: https://abc/auth will become https://abc/auth?requestid=xyz
3) Account requires proof of permission from The Norwegian Data Protection Authority (Datatilsynet)

InitAuth output: InitAuthResponse

Name	Type	Description
StatusID	Int	Status code of the authentication session 4)
StatusText	String	Status of the authentication session
AuthenticationUrl	String	Unique Url to the authentication
RequestID	String	Request Id for this authentication session
TrackingID	String	Tracking Id for this authentication session

4) See 5.1 for a list of possible status codes

4.2 GetAuthStatus

Call this function to receive authentication session status. See 7.2 for sample request.

GetAuthStatus parameters:

Name	Type	Description	Required
DistributorID	String	Provided by Signant	Yes
AccessCode	String	Provided by Signant	Yes
RequestID	String	RequestID from InitAuth	Yes
TrackingID	String	TrackingID from InitAuth	Yes

GetAuthStatus output: GetAuthStatusResponse:

Name	Type	Description
StatusID	Int	Status code of the authentication session 5)
StatusText	String	Status of the authentication session
State	String	State of the authentication session 6)
TrackingID	String	Echo of incoming parameter: TrackingID
UserUniqueID	String	Unique id for this person
UserFullName	String	Full name of authenticated person
UserFirstName	String	First name of authenticated person
UserLastName	String	Last name for authenticated person
UserDOB	String	Date of birth of authenticated person DD.MM.YYYY
UserSSN	String	Social security number of user 7)

5) See 6.1 for a list of possible status codes
6) See 6.2 for a list of possible session states
7) Account requires proof of permission from The Norwegian Data Protection Authority (Datatilsynet)

5 AuthOptions

AuthTypes:

AuthType	Description
NO_BankID	Norwegian BankID, read more at www.bankid.no
NO_BankID_Mobile	Norwegian BankID, read more at www.bankid.no
NO_BuyPass	Norwegian Buypass, read more at www.buypass.no

6 States and status codes

6.1 Status codes

Authentication status code can have the following values:

StatusID	Description
0	OK
2001	Wrong DistributorID and/or AccessCode
2002	Invalid value(s) in options object – see status StatusText for details
4000	General error – see StatusText for details
4001	Invalid or expired RequestID
4002	Wrong RequestID or TrackingID
4003	Session expired

6.2 Session state

Authentication session can have the following states

State	Description
N/A	Session does not exists
NOT_STARTED	Session create but not started
STARTED	Authentication prossess started
INITIALIZED	Authentication provider handshake initialized
PROCESSING	Authentication in process
CANCELED	Authentication canceled by user 8)
FAILED	Authentication failed, see StatusText value for more details
EXPIRED	Session expired
COMPLETED	Authentication completed successfully

8) Cancelled sessions cannot be reused, new session must be requested using InitAuth

7 Sample requests

The sample requests and responses have been shortened down for readability and will therefore not work “as is“, you will have to replace all required fields with correct data before trying to send a request to the service.

7.1 InitAuth sample request

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <InitAuth xmlns="http://tempuri.org/">
      <DistributorID>Your DistributorID</DistributorID>
      <AccessCode>Your AccessCode</AccessCode>
      <AuthOptions xmlns:a="http://schemas.datacontract.org/2004/07/Signant.Auth" xmlns:i="http://www.w3.org/2001/XMLSchema-
instance">
        <a:AuthTypes xmlns:b="http://schemas.datacontract.org/2004/07/Signant.Auth.Enums">
          <b:AuthType>NO_BankID</b:AuthType>
        </a:AuthTypes>
        <a:CancelUrl>https://somesite.com/AuthBegin</a:CancelUrl>
        <a:NextUrl>https://somesite.com/AuthDone</a:NextUrl>
        <a:ReturnSSN>false</a:ReturnSSN>
      </AuthOptions>
    </InitAuth>
  </s:Body>
</s:Envelope>
```

The request above will return the following result:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
  <InitAuthResponse xmlns="http://tempuri.org/">
   <InitAuthResult xmlns:a="http://schemas.datacontract.org/2004/07/Signant.Auth" xmlns:i="http://www.w3.org/2001/XMLSchema-
instance">
    <a:AuthenticationUrl>https://xyz/Init/00000000-0000-0000-0000-000000000000</a:AuthenticationUrl>
    <a:RequestID>00000000-0000-0000-0000-000000000000</a:RequestID>
    <a:StatusID>0</a:StatusID>
    <a:StatusText>OK</a:StatusText>
    <a:TrackingID>11111111-1111-1111-1111-111111111111</a:TrackingID>
   </InitAuthResult>
  </InitAuthResponse>
 </s:Body>
</s:Envelope>
```

7.2 GetAuthStatus sample request

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
  <GetAuthStatus xmlns="http://tempuri.org/">
   <DistributorID>Your DistributorID</DistributorID>
   <AccessCode>Your AccessCode</AccessCode>
   <RequestID>00000000-0000-0000-0000-000000000000</RequestID>
   <TrackingID>11111111-1111-1111-1111-111111111111</TrackingID>
  </GetAuthStatus>
 </s:Body>
</s:Envelope>
```

The request above will return the following result:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
 <s:Body>
  <GetAuthStatusResponse xmlns="http://tempuri.org/">
   <GetAuthStatusResult xmlns:a="http://schemas.datacontract.org/2004/07/Signant.Auth"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
    <a:State>COMPLETED</a:State>
    <a:StatusID>0</a:StatusID>
    <a:StatusText>OK</a:StatusText>
    <a:TrackingID>11111111-1111-1111-1111-111111111111</a:TrackingID>
    <a:UserDOB>1986-06-06T00:00:00</a:UserDOB>
    <a:UserFullName>Ola Nordmann</a:UserFullName>
    <a:UserFirstName>Ola</a:UserFirstName>
    <a:UserLastName>Nordmann</a:UserLastName>
    <a:UserSSN>ddmmyyxxxxx</a:UserSSN>
    <a:UserUniqueID>X-AAAA-BBBB-C-DDDDD</a:UserUniqueId>
   </GetAuthStatusResult>
  </GetAuthStatusResponse>
 </s:Body>
</s:Envelope>
```