Handleiding DigiD koppeling

Stichting Nederland Kennisland, Digitale Pioniers

Datum:

Document naam: digid-zorg-enovatie-handleiding-1.2-publicatie

Versie: 1.2

Inhoudsopgave

1. Inleiding

Dit document is de implementatie handleiding die onstaan is tijdens het maken van de DigiD koppeling voor het Aexist diabetes dossier van de Stichting Zorg-enovatie.

Het grootste gedeelte van dit document is algemeen geldig en toepasbaar. Hoofdstuk 6 beschrijft de oplossing voor het Aexist diabetes dossier van de Stichting Zorg-enovatie en is specifiek voor het daar gebruikte Orbeon platform.

Deze handleiding en de bijbehorende koppeling zijn gemaakt door Open-T in opdracht van Stichting Kennisland.

Aexist diabetes dossier

Het Aexist diabetes dossier van de stichting Zorg-enovatie is een web applicatie die het voor een diabetes patient mogelijk maakt de gegevens van de periodieke metingen en controles die de patient zelf verricht te bewaren en te analyseren.

DigiD

DigiD is het systeem voor digitale identificatie van de Nederlandse overheid. Het is een A-Select authenticatie systeem. Omdat bijna alle Nederlanders beschikken over een DigiD is het voor de hand liggend hier gebruik van te maken voor het Diabetes dossier van de stichting Zorg-enovatie.

2. Technische opzet DigiD

Algemeen

DigiD is een webdienst. De dienst wordt aangeboden via zowel een CGI als een SOAP interface. Het principe van DigiD is in beide gevallen hetzelfde.

Inlogprocedure

Het inloggen met DigiD gaat als volgt:

De web applicatie vraagt een authenticatie sessie aan. Hiertoe wordt een webdienst van DigiD aangeroepen (“AuthenticatieInitiatie').

De applicatie geeft de volgende gegevens door aan DigiD:

- a-select-server
- app_id
- app_url

De a-select-server en app_id zijn van te voren vastgelegd door Logius. De waarde van app_url is de url waar de gebruiker naar toe moet worden gestuurd na afloop van de identificatie.

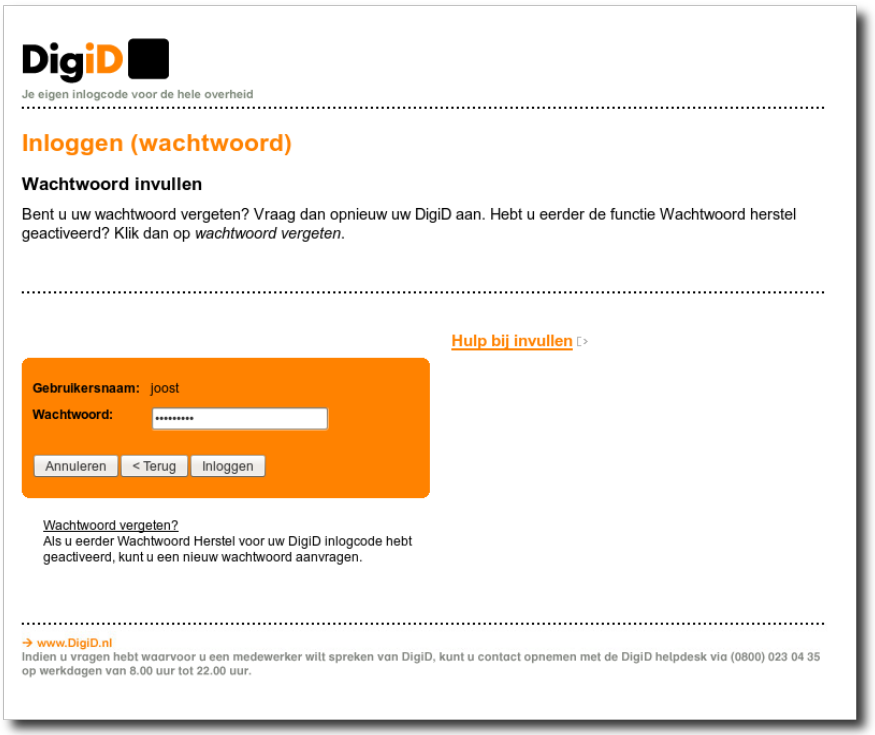
DigiD beantwoordt deze vraag met de volgende gegevens:

- a-select server
- rid
- as_url
- result_code

De eerste 3 gegevens worden gebruikt om een url te construeren waar de gebruiker naartoe moet worden geleid.

De gebruiker krijgt een pagina met uitleg te zien. Door op het DigiD logo te klikken wordt de gebruiker naar de genoemde url geleid.

De gebruiker logt in op de site van digid, met behulp van de volgende 2 schermen:



De browser wordt teruggeleid naar de eerder opgegeven url met de volgende parameters toegevoegd:

- aselect_credentials
- rid
- a-select-server

De applicatie moet nu controleren of de gebruiker werkelijk is ingelogd. Dit gebeurt met de AuthenticatieVerificatie operatie. Deze heeft de volgende parameters:

- a-select-server
- rid
- aselect_credentials

Deze gegevens worden overgenomen van de redirect.

DigiD antwoordt met:

- sector code
- sectoraal nummer
- betrouwbaarheidsniveau
- rid
- organisatie naam
- applicatie id
- resultaatcode
- a-select-server

Als de resultaatcode 0000 is, dan is de gebruiker correct ingelogd. Andere codes duiden er op dat de gebruiker niet goed is ingelogd, de operatie heeft geannuleerd etc.

De 'landing pagina' ziet er in onze testomgeving als volgt uit:

SOAP interface

Logius biedt zowel een CGI als een SOAP interface voor DigiD aan.

Wij zijn uitgegaan van de SOAP interface. Een SOAP interface bestaat uit XML berichten verkeer over een HTTP interface. De berichten worden beschreven in een WSDL welke weer gebruik maakt van XML Schema. Er is uitstekende open source tooling beschikbaar (soapUI) waarmee met dit berichtenverkeer ge-experimenteerd kan worden alvorens tot ontwikkeling van software over te gaan.

3. Technische opzet Aexist diabetes dossier

Het Aexist diabetes dossier is een web applicatie die is opgebouwd met de open source Orbeon formulieren omgeving. Orbeon is een server-side implementatie van de w3c XForms standaard.

Een XForm produceert XML gegevens en kan XML gegevens ontvangen. Orbeon maakt gebruik van XML pipelines om deze gegevens aan te leveren en te verwerken. Binnen een XML pipeline kunnen XML processors gebruikt worden die bijvoorbeeld een SOAP call of een XSLT transformatie uitvoeren.

4. Het inrichten van de pre-productie omgeving

De benodigde documentatie is te vinden op publicatieplein (www.publicatieplein.nl).

Om gebruik te kunnen maken van publicatieplein is een account op de website nodig. Deze site maakt zelf geen gebruik van DigiD.

Na aanmelding op de website moet een geheimhoudingsverklaring moet worden getekend, zie http://www.logius.nl/producten/toegang/digid/aansluiten/

Dit formulier kan niet zomaar gedownload worden. Hiervoor moet contact opgenomen worden met Logius. Hierna kan het (per post) naar Logius verstuurd worden.

Van de publicatieplein website kan vervolgens het formulier 'Aansluitformulier pre-productie omgeving' gedownload worden. Na het invullen en inzenden van dit formulier verstuurt Logius de aansluitgegevens per aangetekende post.

Als u gebruik wilt maken van de SOAP koppeling ontbreekt nog een en ander:

- Documentatie van het SOAP koppelvlak
- SOAP endpoints
- SSL certificaten voor de tweezijdige SSL beveiliging

Deze heb ik door telefoon- en e-mail verkeer met Logius te pakken gekregen. Wij waren de tweede die van dit koppelvlak gebruik maakt en hierdoor loopt een en ander nog niet op de reguliere manier.

5. Het inrichten van de ontwikkel omgeving

Browser certificaat

De allereerste test is het eenvoudigst met de browser, om simpelweg te zien of alle certificaten correct zijn en er verbinding met de preproductie omgeving van DigiD mogelijk is.

Als u een browser laat wijzen naar de WSDL van de DigiD omgeving (https://was-preprod1.digid.nl/was/services/WSDigiDSectorAuthenticatiePortType?wsdl) meldt de browser “Secure connection failed” – SSL peer was unable to negotiate an acceptable set of security parameters (Error code: ssl_error_handshake_failure_alert).

Dit komt omdat er een client-side certificaat met private key nodig is waarmee DigiD vaststelt van welke partij de aanvraag komt. Deze wordt door Logius uitgereikt in de vorm van een PKCS#12 keystore.

De inhoud van de keystore kan getoond worden met behulp van openssl:

```
openssl pkcs12 info -in [keystore filenaam]
```

Om het PKCS#12 certifcaat in firefox te laden:

- Ga naar: edit → preferences
- Selecteer 'Advanced'
- Selecteer de 'Encryption' tab
- Klik op 'View Certifcates'
- Selecteer de tab 'Your Certificates'
- Klik op 'import...'

soapUI

Met soapUI kunnen SOAP calls worden uitgevoerd. SoapUI is in staat om met behulp van de WSDL van de SOAP service vraag berichten te genereren, deze aan de service te versturen en de antwoorden te tonen. Het is ook mogelijk om de omgekeerde weg te bewandelen zodat soapUI zich gedraagt als de webservice die door de WSDL gedefinieerd wordt. Beide functies zijn erg handig bij het werken met een SOAP service.

Om een project in soapUI te starten lezen we de WSDL van de webservice in. Deze WSDL is beschikbaar op de eerder genoemde URL. Echter, de WSDL importeert een XML Schema dat niet beschikbaar is op de aangegeven plaats. We moeten de WSDL dus van een lokale file installeren, en zorgen dat het XML Schema op de correcte plaats aanwezig is.

Dit gaat als volgt:

Aanmaken project en importeren van de WSDL

- File → New soapUI Project
- vul projectnaam in (ik ga verder uit van 'digid'), bij 'Initial WSDL' browse naar de WSDL “WSDigiDAuthenticatie_v1.wsdl”. Zorg dat de file “ WSDigiDAuthenticatie_v1.xsd” in dezelfde directory staat.
- Zorg dat 'Create Requests' aangevinkt is.
- De voorbeeld requests woren nu aangemaakt

PKCS#12 keystore installeren in soapUI

- Rechts-klik op het 'digid' project.
- Klik op 'Show Project View'
- Klik op de tab 'Security Configurations'
- Klik op de sub-tab 'Keystores/Certificates'
- Klik op de '+' in de toolbar
- Browse naar het certificaat en lees dit in.

Het uitvoeren van een AuthenticatieInitiatie operatie:

- Dubbel-klik op de gegenereerde 'Request 1' (onder Projects → digid → WSDigiDSectorAuthenticatiePortType → AuthenticatieInitiatie)
- De request verschijnt. Boven in de toolbar verschijnt het endpoint. Corrigeer deze indien deze niet klopt (het endpoint wordt door Logius aangegeven).
- Klik bij properties (links onder) op SSL keystore en selecteer de eerder ingelezen keystore.
- Vul de ontbrekende gegevens in het Request window in (bij elk ontbrekend gegeven staat een '?')
- Druk op de groene 'start' knop. De aanvraag wordt naar DigiD verstuurd en het resultaat verschijnt in het rechter gedeelte van het window.

Sleutel en certificaten voor de Java omgeving installeren

Om gebruik te maken van de sleutel en certificaten in de Java omgeving waar het diabetes dossier gebruik van maakt, moeten deze beschikbaar gesteld worden aan de java runtime omgeving. Dit is specfiek voor Java applicaties. Applicaties die op een andere techniek gebaseerd zijn zullen dit op een andere manier (specifiek voor dat platform) moeten bereiken.

Het importeren van het PKI Overheid CA Overheid en bedrijven certificaat

Om het PKI overheid CA Oveheid en bedrijven certificaat te installeren wordt het keytool programma gebruikt. Deze voert een operatie uit op een keystore die 'cacerts' heet. Deze maakt deel uit van de Java runtime omgeving, bij mij bevindt deze zich in /usr/lib/jvm/java-6-sun/jre/lib/security.

In de volgende commandoregel moet u de paden naar DigiNotarPKIoverheidCAOverheidenBedrijven en cacerts vervangen door de juiste voor uw systeem:

```
keytool -import -v -trustcacerts -alias DigiNotarPKIoverheidCAOverheidenBedrijven -file
~/projects/digid/DigiNotarPKIoverheidCAOverheidenBedrijven -keystore cacerts -storepass
changeit
```

Het meegeven van het client certificaat gaat met behulp van command-line opties van java. Het pad naar de keystore moet vervangen worden door het correcte pad voor uw systeem, en het password door het echte password:

```
-Djavax.net.ssl.keyStoreType=pkcs12
-Djavax.net.ssl.keyStore=~/projects/digid/opent.SDMC.keystore.p12
-Djavax.net.ssl.keyStorePassword=vervangditdoorhetechtekeystorepassword
```

6. Omschrijving van de gemaakte code

Overzicht

Deze sectie beschrijft de code zoals die gerealiseerd is voor Stichting Zorg-enovatie. Deze sectie is dus specifiek voor de daar gebruikte oplossing (Orbeon).

Er zijn een aantal XML pipelines en XForms ingericht om de koppeling met DigiD te realiseren.

Deze is geheel losgekoppeld van de rest van de applicatie, en is eenvoudig opnieuw te gebruiken voor een andere applicatie die ook gebruik maakt van Orbeon XForms.

Nadat de authenticatie procedure is doorlopen wordt een XML document in de web server sessie achtergelaten met hierin het laatste responsebericht van de verificatieslag. De applicatie moet deze sessie variabele lezen en hiermee een gebruiker toelaten of weigeren.

Authenticatie slag (digid-authenticate.xpl)

De XML pipeline wordt aangeroepen als model van de XForm (en wordt dus uitgevoerd voordat de XForm getoond wordt).

Deze pipeline doet simpelweg een SOAP call met de AuthenticatieInitiatie operatie. De output van de pipeline is het gehele SOAP response bericht.

De XForm heeft de beschikking over dit XML bericht en gebruikt dit om de url naar de DigiD inlog pagina te construeren (simpelweg door enkele elementen aan elkaar te zetten met & er tussen).

Verificatie slag (digid-verify.xpl)

De verificatie slag is ook uitgevoerd als een XML pipeline. Deze wordt uitgevoerd voordat de 'verify.xhtml' pagina getoond wordt.

De pipeline voert de volgende stappen uit:

- Haal de parameters van de http GET request op
- Maak een SOAP request bericht met behulp van een XSLT transformatie
- Voer de SOAP call naar de AuthenticatieVerificatie operatie uit
- Plaats het SOAP bericht in de 'digid' sessie variabele
- Geef de gehele SOAP response aan de pagina

Ophalen van de digid sessie en het reageren daarop (digid-session.xpl)

Deze XML pipeline leest de digid sessie. Tevens reageert de pipeline op het in de sessie aanwezige resultaat door een redirect te doen.

De pipeline voert de volgende stappen uit:

- Lees de digid variabele (het oorspronkelijke SOAP response bericht) uit de sessie.
- Als de variabele leeg is wordt de gebruiker naar de /digid/notloggedin pagina gestuurd
- Als de variabele aanwezig is maar de SOAP response is niet 0000 dan wordt de gebruiker naar de /digid/error pagina gestuurd.
- Tenslotte wordt de inhoud van de pagina doorgegeven aan de XForm.

Digid sessie beeindigen (kill-digid-session.xpl)

Deze pipeline maakt simpelweg de inhoud van de digid sessie variabele leeg.

7. Aanbevelingen voor het vervolg bij Stichting Zorg-enovatie

De omgeving is geinstalleerd bij Stichting Zorg-enovatie en functioneert naar behoren. Zodra de hele applicatie gereed is om te testen verdient het wel aanbeveling om de door Logius gehanteerde checklist nog eens na te lopen, omdat deze veel eisen bevat die voor de gehele oplossing gelden (sessie timeouts, verwijzingen naar DigiD etc.) Deze checklist is beschikbaar op de publicatieplein website.

8. Samenvatting van de te nemen stappen voor het aansluiten van een applicatie op DigiD

De bovenstaande beschijving is specifiek voor de Aexist diabetes dossier applicatie van de Stichting Zorg-enovatie. In het algemeen zijn de te nemen stappen:

- Toegang verkrijgen tot Publicatieplein
- Pre-productie aansluiting aanvragen bij Logius
- Het inrichten van een ontwikkel/pre-productie omgeving
- Het verkrijgen en installeren van het PKCS#12 certificaat voor toegang tot de DigiD service
- Het testen van de aansluiting met een generieke SOAP tool (zoals soapUI)
- Het vinden van een geschikte SOAP library om met de DigiD service te communiceren (dit verschilt per platform)
- Het inrichten van de code voor de authenticatie slag
- Het inrichten van de code voor de verificatie slag
- Het inrichten van de bijbehorende formulieren
- Het controleren van de ingerichte applicatie met de checklist van Logius
- Testen van de pre-productie omgeving door Logius
- Daarna herhaalt de inrichting- en test slag zich voor de productie omgeving