How to register **id4me** agents/identities at a foreign **id4me** authority?

Peter Höbel
Senior Developer, Open-Xchange

October 2019

Stay Open.

Agent Identity Management API

https://id.denic.de/aim/docs/api/v1/

- The AIM API provides endpoints to manage agents and identifiers at an authority
- Any request consists of a serialized JSON Web Signature with detached payload
- There are two different kinds of request
  - The request contains a JSON Web Key in the JWS
    - This request is only used to create an new account for an agent
    - The id field from the response is the agentId
  - The request contains no JWK but instead a kid field in the JWS header
    - This request is used for any other operations
    - The kid field in this request means always the agentId

Agent Identity Management API

Notes

- Not all response codes are mentioned in the documentation
  - 403 Forbidden
    - If you try to create a new identity with an already registered identifier
  - 410 Gone
    - If you try to open the “magic url” a second time for example.
  - 412 Precondition Failed
    - This response means that the ACME record in the DNS is not found. We should retry this call later.
  - The current status of an identity cannot be determined
    - You have to track your registrations by yourself
    - This will be possible in future versions of the api

Steps to create an identity

Inclusive agent registration

1. Create a new account for an agent at the authority
    - Save the response body because we need at least the field id as agentId
2. Create a new identity authorization
    - Save the response body because we need the id for further requests
3. Manually add the acme record to your DNS zone
    - _acme-challenge.hans 300 IN TXT "4V3Vc1NQf0sddfxS7MtMAcCmjQO5vAJdAZA5N_BoWqk"
4. If not exists, add the discovery reccord to your DNS zone
    - _openid 3600 IN TXT "v=OID1;iss=id.staging.denic.de;clp=wso.open-xchange.com"
5. Notify the authority that DNS setup has been completed
6. Open the magic url and set up a password

Java AIM wrapper API

Java api wrapper

- Encapsulates all requests of the AIM api
- Each call returns
    - Status code
    - Status message
    - Response body (if available)
- Can be imported into a java project as maven dependency
- ```
  <dependency>
      <version>2.0</version>
      <groupId>org.id4me</groupId>
      <artifactId>aim-api-wrapper</artifactId>
  </dependency>
  ```
  ```

Spring Boot demo application

Overview

- Consists essentaially of three controllers and one authorization filter
- ID4meRelyingPartyFilter
  - Ensures that some request require an id4me logon
- ID4meAimAgentController
  - Handles any agent related requests
- ID4meAimAuthorizationController
  - Handles any identity related requests
- ID4meUserinfoController
  - Endpoint for the /userinfo request of local identities

Spring Boot demo application

Configuration

- Any configuration parameters are in application.properties

  You can specify the location of the file with a parameter

  java -jar -Dspring.config.location=/opt/id4me/application.properties id4me-identity-agent-1.0.0-SNAPSHOT.jar

- You need a private/public key pair in PEM format
- You need a jwks.json file
  - If no jwks.json file is found it will be created at server start
- The local data is stored in file system
  - The application must have read/write access to the data path
  - To change this you can create your own org.id4me.impl.ID4meDataImpl.class

Spring Boot demo application

Sample application.properties

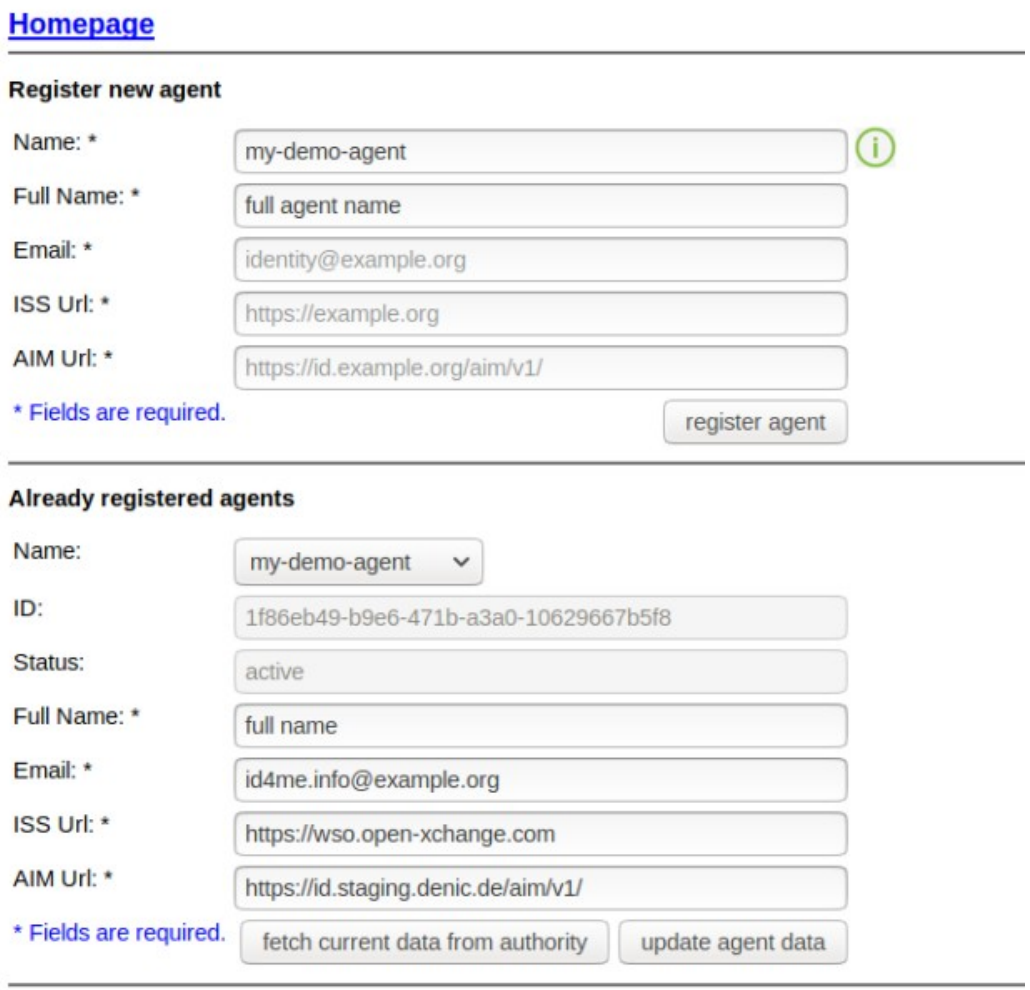
```
# Comma separated identifiers of administrators.
id4me.admins=admin.example.org
# A local file path with read/write access for ID4meDataImpl.class
id4me.data.path=/opt/id4me
# A private key file in PEM format. Used by ID4meDataImpl.class
id4me.private.key=/opt/id4me/id4me.priv.pem
# A public key file in PEM format. Used by ID4meDataImpl.class
id4me.public.key=/opt/id4me/id4me.pub.pem
# Path to jwks.json. The file will be created if not found.
jwks.json=/opt/id4me/jwks.json
# The iss for ./well-known/openid-configurationn attributes issuer,jwks_uri, userinfo_endpoint.
id4me.iss=https://example.org
# Comma separated list of urls with anonymous access. DO NOT CHANGE WITHOUT SPECIFIC REASON!
id4me.excluded.urls=/userinfo,/.well-known/openid-configuration,/openid-configuration,/jwks.json,/,/favicon.ico,/js/*.*,/img/*.*,/css/*.*,/identity.html,/agent/
list,/identities/create,/authz/*.*
# id4me properties values for the relying party filter configuration
registration.data.path=/opt/id4me/registrations
redirect.uri=https\://example.org/logon
dnsssec_root_key=. IN DS 20326 8 2 E06D44B80B8F1D39A95C0B0D7C65D08458E880409BBC683457104237C7F8EC8D
client.name=Claims-Provider-Demo
client.max_fetch_size=50000
dns.resolver=8.8.8.8
dnssec_required=false
scopes_fallback=true
```

Screenshots

https://id4me.tismail.de/

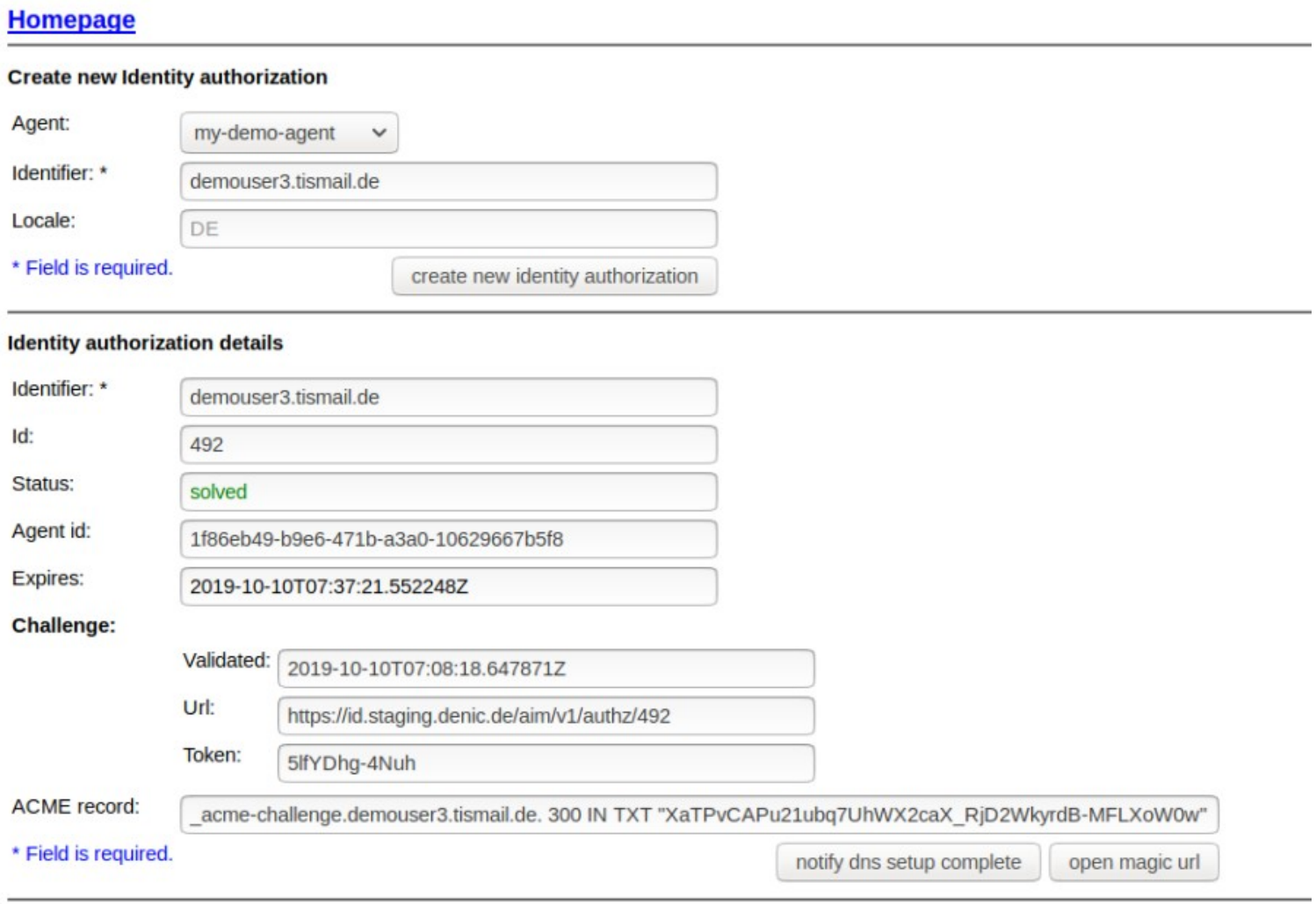
Register a new agent

https://id4me.tismail.de/agent.html



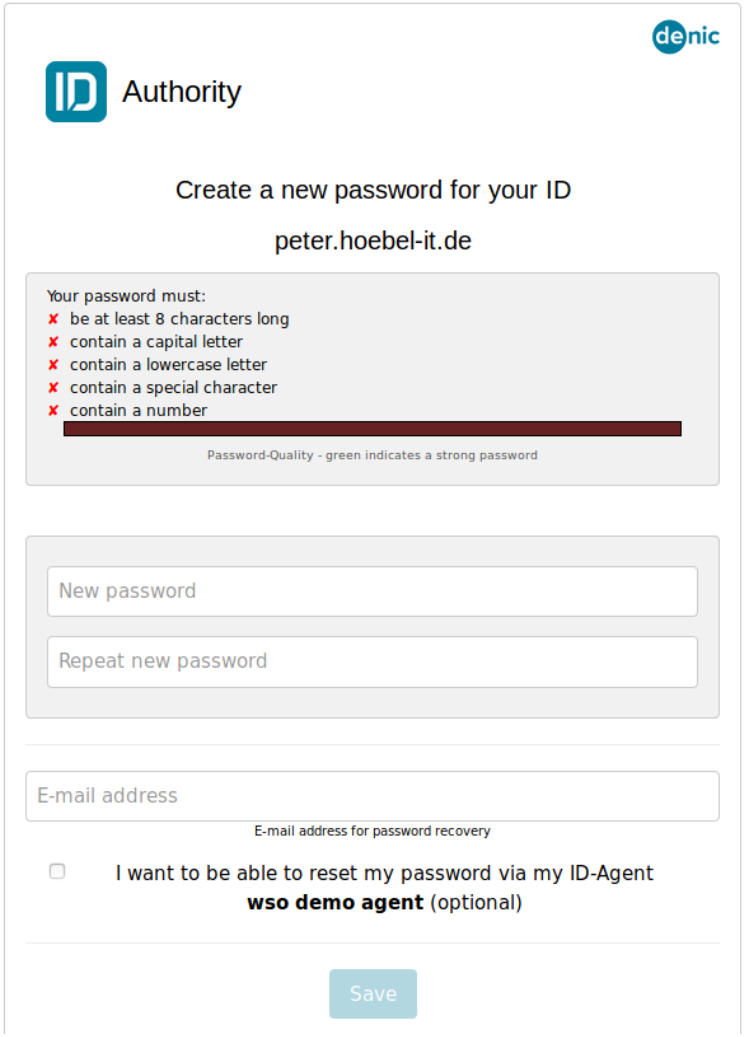
Create a new identity

https://wso.open-xchange.com/identity.html



Open the magic url

https://id.staging.denic.de/init/setup/?magic=cAZcPYkGztYv01ToTrsbcNeTaxQLXn4dhwrJWEZzOk

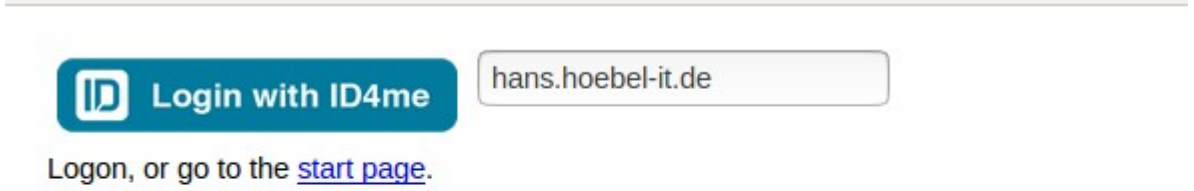


Edit userinfo data

http://wso.open-xchange.com/userinfo.html

...and finally log on

http://wso.open-xchange.com/logon.html



Thank you!

Application at gitlab

Thank you very much for your attention!

The source code of the project is available at our gitlab:
git clone https://gitlab.com/ID4me/id4meidentityagent.git

Open-Xchange AG

Rollnerstraße 14
D-90408 Nuernberg

Phone: +49 2761-8385-0
Fax: +49 2761-8385-30

info@open-xchange.com
www.open-xchange.com

Stay Open.