collecting_society - Schnittstellen #365

Add remote authentication and password change/reset for c3s members via c3sMembership api

04/17/2017 04:40 AM - Alexander Blum

Status:	Neu	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Sarah Stoffels	**% Done:**	0%
Category:		**Estimated time:**	8.00 hours
Target version:	Repertoire 6) Post Production	**Spent time:**	0.00 hour

Description

C3S members should be authenticated remotely via the c3sMemberhip api.
This creates also the need for changing/resetting the password remotely.
Non C3S members should still be authenticated via repertoire db.

The following descriptions needs a bit more specification coordinated with Markus.

Remote authentication

- on member login, email and password (hash?) is transmitted to membership
- depending on the response, the user is logged in or denied

Password change

- password change form: email, old/new/confirmed password
- request (password hashes?) is relayed to membership
- confirmation/errors are fed back to the user

Password reset

- password forgotten form: email
  - request is relayed to membership, where a token is generated and included in the response
  - if the response is ok (email belongs to a member, etc.) a mail is sent to the web user via repertoire with a URL containing email an generated token
- form reset view for the URL, e.g. /reset-password/EMAIL/TOKEN
  - request is relayed to membership
  - if email/token matches
    - the user is logged in
    - user feedback via flash message
  - if email/token does not match
    - redirect to /
    - minimal user feedback

Related issues:

Related to collecting_society - Schnittstellen #361: Update c3sMembership api...	**Neu**

History

#1 - 04/17/2017 04:40 AM - Alexander Blum

- Related to Schnittstellen #361: Update c3sMembership api service added

#2 - 06/27/2018 11:29 PM - Alexander Blum

- Target version deleted (2) Testing phase II)

postponed until after production phase

#3 - 11/05/2018 03:01 PM - Alexander Blum

- Target version set to 6) Post Production

#4 - 10/08/2019 03:42 PM - Alexander Blum

- Target version changed from 6) Post Production to Repertoire 6) Post Production

#5 - 10/08/2019 03:50 PM - Alexander Blum

- Project changed from repertoire to collecting_society

#6 - 10/11/2019 01:54 PM - Markus Lorenz

Alexander Blum wrote:

> C3S members should be authenticated remotely via the c3sMemberhip api.
> This creates also the need for changing/resetting the password remotely.
> Non C3S members should still be authenticated via repertoire db.

Don't you think we should rather have a central authentication system used by other services like Repertoire and Membership? Splitting this up seems to make the architecture and logic quite complicated.

> The following descriptions needs a bit more specification coordinated with Markus.
>
> **Remote authentication**
>
> - on member login, email and password (hash?) is transmitted to membership

Password or hash decides about who does the authentication and has the corresponding responsibility. If Membership delivers a password hash then it does not do the authentication. In this case Repertoire does the password validation, therefore authentication and thus takes responsibility. Splitting this across multiple services feels like an anti-pattern.

> - depending on the response, the user is logged in or denied
>
> **Password change**
>
> - password change form: email, old/new/confirmed password
> - request (password hashes?) is relayed to membership
> - confirmation/errors are fed back to the user
>
> **Password reset**
>
> - password forgotten form: email
>   - request is relayed to membership, where a token is generated and included in the response

What kind of token is this and what is it used for?

> * if the response is ok (email belongs to a member, etc.) a mail is sent to the web user via repertoire with a URL containing email an generated token

There's again a lot of mix up of responsibilities. If Membership stores the login information – which it currently does for some members but I'm not sure it should in a wide services ecosystem – then why is Repertoire sending password reset emails? We need keep functionality cohesive in separate services and not spread it across multiple of those.

> - form reset view for the URL, e.g. /reset-password/EMAIL/TOKEN
>   - request is relayed to membership
>   - if email/token matches
>   - the user is logged in
>   - user feedback via flash message
>   - if email/token does not match
>   - redirect to /
>   - minimal user feedback

#7 - 10/28/2019 09:53 PM - Markus Lorenz

In the past we had discussions about how services should authenticate towards each other. I just came across a mechanism which seems to be well established and which we might want to adapgt: JWT (JSON Web Tokens) [1]. There's a Python implementation called PyJWT [2].

I found it via privacyIDEA [3], a 2 Factor Authentication system, which is using JWT to communicate with other services using the 2FA feature of privacyIDEA.

[1] https://blog.logrocket.com/how-to-secure-a-rest-api-using-jwt-7efd83e71432/
[2] https://pyjwt.readthedocs.io/en/latest/
[3] https://www.privacyidea.org/about/features/