collecting_society - Webfrontend #392

Add dashboard content

04/17/2017 05:59 AM - Alexander Blum

Status:	Erledigt		**Start date:**	
Priority:	Hoch		**Due date:**	
Assignee:	Sarah Stoffels		**% Done:**	100%
Category:			**Estimated time:**	1.00 hour
Target version:	Repertoire 1) Testing phase I		**Spent time:**	34.50 hours

Description

- dummy intro text
- dummy intro picture
- important objects
  - content with errors
  - orphaned content (unassinged to a creation)
  - uncommited metadata objects

Associated revisions

Revision 0367d530 - 03/03/2018 03:00 PM - sarah-github

Merge branch 'dashboard-#392' into develop

Revision 0367d530 - 03/03/2018 03:00 PM - Sarah Stoffels

Merge branch 'dashboard-#392' into develop

Revision 2a344ba7 - 03/05/2018 05:59 PM - Alexander Blum

ref #392: fixes complexity of object instances in registry

Revision 2a344ba7 - 03/05/2018 05:59 PM - Alexander Blum

ref #392: fixes complexity of object instances in registry

History

#1 - 04/17/2017 09:06 PM - Alexander Blum

- Description updated

#2 - 01/21/2018 02:00 PM - Sarah Stoffels

- Assignee set to Sarah Stoffels

#3 - 02/07/2018 04:02 PM - Sarah Stoffels

- Status changed from Neu to In Bearbeitung

- % Done changed from 0 to 10

because frontend didn't work and ressources on laptop are too slow when rebuilding everything (6./7.02.2018)

#4 - 02/07/2018 04:19 PM - Sarah Stoffels

in ado/src/collecting_society.portal.repertoire/collecting_society_portal_repertoire/templates/repository/dashboard.pt

#5 - 02/08/2018 02:29 PM - Sarah Stoffels

- % Done changed from 10 to 30

7.5h, da zu dritt 2.5h

#6 - 02/18/2018 01:19 PM - Alexander Blum

- Description updated

change of 'duplicates' to more general 'rejected' content (name not defined yet, should be short and easily understandable). content might have numerous rejection reasons (dupliacte, quality, ...), wich all have the same workflow (deletion after ~7 days, user should be able to stop deletion by starting a dispute).

new widget "orphaned content" (name also not defined yet), which is important for the user to track the "todos". Content objects without a relation don't have any purpose in our system.

#7 - 02/18/2018 02:25 PM - Alexander Blum

I replaced Content.user with the entity creation workflow (entity_origin, entity_creator). Further I changed the entity_creator foreign table to party.party instead of res.user. This might brake your code. For the changes, see

- DB: https://github.com/C3S/collecting_society/commit/9765849aa7faaf04f70fad1e248bb13fa46586be
- Model Wrapper: https://github.com/C3S/collecting_society.portal.creative/commit/d3d1131be5836af9172571e56ef79ca7a0e0036a

#8 - 03/05/2018 01:59 AM - Sarah Stoffels

- Status changed from In Bearbeitung to Erledigt

- % Done changed from 30 to 100

learned a lot of context about registry, views, tryton data-model and templates here (causing the amount of spent time).

#9 - 03/05/2018 05:49 PM - Alexander Blum

One need to be cautious with putting objects within the registry, as deepcopies are made to merge the tree of registry nodes into the final registry. If instances of classes with complex things in it (like the whole request or references to tryton models) need to be put into the registry, there's a possibility to define a function **deepcopy** within the class, which is called on a deepcopy and should return the new object.

By saving the whole request into an object put into the registry, a recursion is triggered, as request.context contains the registry again, which includes the object containing the registry, ect.

For now, I defined a new request method (request.party) to easily retrieve the current party from any request and saved only the request.party.id within the instance, circumventing the problem.

#10 - 10/08/2019 03:34 PM - Alexander Blum

- Target version changed from 1) Testing phase I to Repertoire 1) Testing phase I

#11 - 10/08/2019 03:51 PM - Alexander Blum

- Project changed from repertoire to collecting_society