

collecting_society - Konzept #324

Should objects of others be editable, if you have created it?

04/11/2017 10:36 PM - Alexander Blum

Status:	Erledigt	Start date:	
Priority:	Hoch	Due date:	
Assignee:	Alexander Blum	% Done:	0%
Category:		Estimated time:	1.00 hour
Target version:	Repertoire 1) Testing phase I	Spent time:	0.00 hour
Description			
Usecase 1			
The web user adds a group artist for his band of 4 members. He is the only registered artist, so he needs to create the other 3 artists (name, email)			
Should the web user be able to edit the created artists? To what extent?			
Usecase 2			
The web user adds a creation. He needs to create an artist, who wrote the lyrics for this song. He needs to create three creations of other artists, as they do not exist.			
<ul style="list-style-type: none">• The first artist exists in our database and is claimed.• The second artist exists in our database and is unclaimed.• The third artist does not exist in our database.			
Should the web user be able to add the creation for the first artist?			
Should the web user be able to edit the first/second/third created artist?			
Related issues:			
Related to collecting_society - Konzept #270: Behaviour after user deleted an...		Erledigt	
Related to collecting_society - Konzept #340: GUI to create foreign objects		Erledigt	
Blocked by collecting_society - Konzept #341: Creation/Edit of foreign creations		Erledigt	

History

#1 - 04/11/2017 10:37 PM - Alexander Blum

- Related to Konzept #270: Behaviour after user deleted an object? added

#2 - 04/12/2017 09:10 PM - Meik Michalke

Usecase 1

Should the web user be able to edit the created artists? To what extent?

as long as that artist is not claimed, he should be able to edit name and mail address, e.g. to correct typos. but anything else should only be editable by someone who claimed it.

Usecase 2

Should the web user be able to add the creation for the first artist?

yes, but he should leave it to the artist to commit -- is that possible? that way, the original artist could check the validity and still decide to discard the entry without raising a dispute.

Should the web user be able to edit the first/second/third created artist?

first: no way

second/third: only if he created the artist, then treat it like usecase 1. if someone else created the second case, only that one should be in the position of usecase 1.

#3 - 04/12/2017 09:10 PM - Meik Michalke

- Assignee changed from Meik Michalke to Alexander Blum

#4 - 04/16/2017 06:15 AM - Alexander Blum

- Assignee changed from Alexander Blum to Meik Michalke

- Estimated time set to 1.00

I added a description to the [Specification](#).

as long as that artist is not claimed, he should be able to edit name and mail address, e.g. to correct typos. but anything else should only be editable by someone who claimed it.

This could work in principle and it's implementation depends on the solution of [#340](#). It might need a hand crafted form (~3h).

We also need to provide the user with information, why an object created by him is not editable anymore - e.g. by mouseover on an deactivated edit icon.

Should the web user be able to add the creation for the first artist?

yes, but he should leave it to the artist to commit -- is that possible? that way, the original artist could check the validity and still decide to discard the entry without raising a dispute.

The object is auto committed and should be - all related objects must not be uncommitted, as they could be a distribution relevant part of the system (being the reason for the commit workflow). In this case, some Artist X referenced one of his Creations A as a derivative of the Creation B of another Artist Y.

Should Y really be able to discard B? The normal usecase for the need of a discard would be the claim of an artist with a wrong collection. Perhaps there should be a "unclaim" button/mechanism for Releases and Creations (but not Artists), by which the creation/release(+creations) is split from the artist and added to another new artist with the same name?

To implement, what you have in mind, we should expand the claim process by a "claim_revised" flag. Unrevised objects could then also be promoted on the dashboard distinguished from the uncommitted ones. The question then is, if changes to unrevised objects should be handled exactly like committed objects - those objects are related and we could loose information, if a freshly claimed artist would change them.

Should the web user be able to edit the first/second/third created artist?

second/third: only if he created the artist, then treat it like usecase 1. if someone else created the second case, only that one should be in the position of usecase 1.

Thinking it through, the solution depends on how to relate to Artists during the creation of a foreign Creation, see [#341](#). If we decide to omit the search for artists in the database and don't want to match Artists just by name, the first and second case is not possible.

#5 - 04/16/2017 06:17 AM - Alexander Blum

- Related to Konzept #340: GUI to create foreign objects added

#6 - 04/16/2017 06:18 AM - Alexander Blum

- Blocked by Konzept #341: Creation/Edit of foreign creations added

#7 - 04/16/2017 06:36 AM - Alexander Blum

Maybe an unclaim of an artist should also be possible?

#8 - 04/16/2017 08:54 AM - Alexander Blum

as long as [...] not claimed, he should be able to edit

Shouldn't the same rules for [changes to committed objects](#) apply to changes to foreign objects? It seems unintuitive, that e.g. original/derivative Creations may not be added/removed but changed without a dispute.

#9 - 04/16/2017 09:35 AM - Alexander Blum

"unrevised" should not be a flag on it's own, but an additional state in the claim workflow.

#10 - 05/01/2017 03:26 PM - Meik Michalke

- Assignee changed from Meik Michalke to Alexander Blum

In this case, some Artist X referenced one of his Creations A as a derivative of the Creation B of another Artist Y.

Should Y really be able to discard B?

i think there's two possible cases here:

1. if B existed before in the database and Y had already claimed it, Y should be able to discard the relation between A and B
2. if B didn't exist before but was newly created by X, Y should be able to reject B completely

i assume that Y would only do that if B is indeed not his/her creation or if he/she knows for sure that A is not a derivative of it. that is because Y wouldn't gain anything by rejecting, only by accepting the link, so there's no incentive for falsely rejecting relations from Y's perspective.

the question is what should happen to B if Y rejects the creation. if Y doesn't want to claim it, it should probably be kept as a creation by an unclaimed artists with the same name, like you proposed, because there's also no incentive for X to falsely adding such a relation (because it will split the revenue). in cases where there are other artists by the same name already, they should be able to directly see the unclaimed work somewhere in their profile, so they can check if it's theirs and then probably claim it.

#11 - 05/27/2017 08:05 PM - Alexander Blum

- Assignee changed from Alexander Blum to Meik Michalke

1. if B existed before in the database and Y had already claimed it, Y should be able to discard the relation between A and B

This could be covered by the commit workflow I proposed concerning the revision of freshly claimed objects. My question was, if those should be handled exactly like uncommitted objects, so everything (including original creations) could be changed without dispute until revised. after revision, no easy changes anymore, but a dispute over why to change some metadata.

In general this sounds reasonable, as the user should be provided with an easy edit phase after claiming a bunch of auto-created objects. On the other hand, this could possibly be an attack vector, as someone could claim all unclaimed songs and change all the metadata. And those auto-created objects could already have been used for money distribution.

1. if B didn't exist before but was newly created by X, Y should be able to reject B completely

I am with you, that there's no incentive for both sides to manipulate the data. But I disagree in general, that any author should be able to reject derivatives - those are declared derivatives by their artists. If you have a decent popular hit, you probably cannot possibly validate all derivatives. In short, information about the derivatives is not information of the original artist, but of each derivative artist and therefore should not be perturbed by the original author.

By the same reasoning, I propose to omit the definition of derivative creations at all, and to just provide the original creations see [#399](#).

#12 - 05/29/2017 11:16 AM - Meik Michalke

attack vector

attack should be alarmed ;-)

In general this sounds reasonable, as the user should be provided with an easy edit phase after claiming a bunch of auto-created objects.

but in general it's not web users who should remain in control of the database, but us. and we need to provide working mechanisms against fraud, even if it means more work. at a later point we might introduce mechanisms to allow some mere error corrections (like heuristics to detect typo

corrections in work titles, but not author names), but for a first phase it should just be accept or dispute.

But I disagree in general, that any author should be able to reject derivatives

oh, but s/he should, see [#399](#).

#13 - 05/29/2017 11:16 AM - Meik Michalke

- Assignee changed from Meik Michalke to Alexander Blum

#14 - 05/29/2017 07:10 PM - Alexander Blum

- Status changed from Feedback to Erledigt

Ok.

#15 - 10/08/2019 03:34 PM - Alexander Blum

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

#16 - 10/08/2019 03:50 PM - Alexander Blum

- Project changed from repertoire to collecting_society