

Allow setting issue assignment by username via REST interface

2022/05/09 15:06 - Admin Redmine

ステータス:	New	開始日:	2010/10/21
優先度:	通常	期日:	
担当者:		進捗率:	0%
カテゴリ:	REST API_32	予定工数:	0.00時間
対象バージョン:		作業時間:	0.00時間
Redmineorg_URL:	https://www.redmine.org/issues/6721	status_id:	1
category_id:	32	tracker_id:	2
version_id:	0	plus1:	0
issue_org_id:	6721	affected_version:	
author_id:	21120	closed_on:	
assigned_to_id:	0	affected_version_id:	
comments:	19		
説明			
<p>h2. Background</p> <p>The REST interface allows setting the assigned_to_id on an issue. However, this requires that the user's ID number is known - currently this is unavailable. Even when this becomes available, discovering the ID would require a second HTTP request.</p> <p>h2. Proposed Feature</p> <p>Allow sending the assigned_to_name field through both the POST (create new issue) and PUT (update issue) operations.</p>			
journals			
<p>this feature would be VERY useful for my application. I currently have to use "Admin" user to create tasks, which is a bad solution</p> <p>without this feature, tasks manipulation via REST is quite limited. say, I want to load data from Redmine, edit assignments in some other program (Microsoft Project in my case) and then save back to Redmine. I don't know user database IDs (doh!), but Redmine expects "assigned_to_id" parameter set for issues. This is definitely not user-friendly. The assignment should be possible via user login name.</p> <p>I have created a patch for this feature against the trunk (at revision 8513). Attached is the diff file.</p> <p>Thank you, Sridhar. I really hope this will be accepted by Redmine guys...</p> <p>This will work only when the "assigned_to_id" is NOT set and "assigned_to_name" is set. To be more specific, this will work only when params[:issue][:assigned_to_id] is NOT set and params[:issue][:assigned_to_name] is set.</p>			
Alexey Skor wrote:			
<p>Thank you, Sridhar. I really hope this will be accepted by Redmine guys...</p>			
The problem with this patch is that it only works for issue creation/update.			
Maybe there should be a more specific solution; doesn't the @project_id@ parameter accept the project natural key?			

Etienne Massip wrote:

The problem with this patch is that it only works for issue creation/update.

Maybe there should be a more specific solution; doesn't the @project_id@ parameter accept the project natural key?

Etienne: What are the other methods that need the assigned_to_name feature? While creating the feature, Ian Epperson wrote that he needed this feature for create and update.

For consistency this could include @GET /users/[username].:format@, @GET /issues.:format?assigned_to_id=[username]@, @POST/PUT news@, etc.

More functions are available with the login name now.

show, edit, update, destroy, edit_membership and destroy_membership of the users controller are available with the login name now.

@GET /users/[username]

GET /users/[username]/edit

PUT /users/[username]

DELETE /users/[username]

PUT /users/[username]/memberships/[membership-id]

DELETE /users/[username]/memberships/[membership-id]

@

NOTE: The functions for the users controller will fail on the edge case where the login-name is composed of digits only.

Issues can now be accessed with the "assigned_to_name" parameter;

@GET /issues.:format?assigned_to_name=[username]@

News table does not have the assigned_to_id field. It has the author_id which defaults to the currently logged in user.

Attached is the patch.

I was wondering if the @Project@ model way could be applied:

source:/trunk/app/models/project.rb#L253

Etienne Massip wrote:

I was wondering if the @Project@ model way could be applied: source:/trunk/app/models/project.rb#L253

As I understand it, yes, it can be done. That would impact every User.find([id]) call. If that is the preferred way of doing, then I will do it. Can you confirm that it is the preferred way?

Here you go. User.find works the same way that Project.find does; you can get user by User.find([username]) and User.find([id]).

I have used better names for the functions that fetch the id from the login name.

Sridhar P wrote:

Here you go. User.find works the same way that Project.find does; you can get user by User.find([username]) and User.find([id]).

Yes, I think it's cleaner now since the behavior will be the same everywhere in the app.

I have used better names for the functions that fetch the id from the login name.

Actually you should remove the changes in @IssueController@ and use the @assigned_to_id=@ syntax (for consistency with @project_id@).

Etienne Massip wrote:

Actually you should remove the changes in @IssueController@ and use the @assigned_to_id=@ syntax (for consistency with @project_id@).

As far as I know project_id is cannot be updated with the api. So, it is in a different league.

The "assigned_to_id" field can be updated. If we allow the login name to be sent as a value for the "assigned_to_id" key, we'll have to modify the assigned_to_id and set it to the id of the user if we encounter a login name. Is that okay?

Also, this will have same edge case scenario as with the users model. If a user has a login name composed of digits only, then we'll have the same problem (login name will be considered to be an id, because it contains digits only).

Do you still want me to go ahead with the @assigned_to_id=[login]@ syntax?

I tried to do it but it is interfering with other methods:

- Create issue with group assignment
- Update issues with group assignment

These methods use the group id for the assigned_to_id field. @assigned_to_name@ seems to be a safe bet.

Looks like I had an unrelated error. I have modified the code to enable the usage of @assigned_to_id=[login]@. You should be able to use login instead of id.

can we just keep "assigned_to_id" for backward compatibility and add "assigned_to" parameter, which would take either ID or Login name? if "assigned_to" is used, then "assigned_to_id" is ignored (and should be marked as "deprecated" and removed in the future Redmine releases).

using "assigned_to_id" for both "id" and "login" seems like a bad decision to me.

if the login name contains numbers only, then this user is plain stupid and we should ignore or punish him =:-] just kidding. we can disable digits-only login names if we want to reuse the same "ID or String Key (which is a login name in this particular case)" approach in the other parts of the API to make it more consistent.

Given a choice I would stick with @assigned_to_name@. The intent is clear and it is unambiguous. But then the decision isn't in my hands :)

related_issues

duplicates,Closed,8531,Task's "assignee" should be specified by login instead of user ID

履歴

#1 - 2022/05/10 17:21 - Admin Redmine

- カテゴリ を REST API_32 にセット