

## Project filter is (sometimes) ignored in CSV export

2022/05/09 18:31 - Admin Redmine

|                 |                                      |                      |            |
|-----------------|--------------------------------------|----------------------|------------|
| ステータス:          | New                                  | 開始日:                 | 2022/05/09 |
| 優先度:            | 通常                                   | 期日:                  |            |
| 担当者:            |                                      | 進捗率:                 | 0%         |
| カテゴリ:           | Issues filter_56                     | 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/28669 | status_id:           | 1          |
| category_id:    | 56                                   | tracker_id:          | 1          |
| version_id:     | 0                                    | plus1:               | 0          |
| issue_org_id:   | 28669                                | affected_version:    |            |
| author_id:      | 40856                                | closed_on:           |            |
| assigned_to_id: | 0                                    | affected_version_id: | 139        |
| comments:       | 0                                    |                      |            |

**説明**

Suppose you have a project P which has subprojects P1. Further you have a global, saved issue query with a @project\_id@ filter for just the subproject P1.

When calling that query in the context of P (by going to /projects/p/issues?query\_id=123), only issues of P1 are shown. The same is true for the PDF export when accessed from this page. However, the CSV export when accessed from the same page, will contain issues from P as well.

The cause of that discrepancy is, that the modal CSV export options window serializes the query into a bunch of hidden fields, while the PDF export just uses the @query\_id@ URL parameter, like the web UI view does.

Deeper down, the problem is the timing of when @@query.project@ is set in @QueriesHelper#retrieve\_query@: With @query\_id@ present, the query is loaded and then the project is set. Without @query\_id@ (in the CSV export case), the project is set first, before the params are applied. This results in a different available\_filters configuration, because only in absence of the project, the @project\_id@ filter will be added. So with the project being set early, the project\_id filter isn't initialized in the query object and thus all issues of P are rendered in the CSV. I'm attaching a failing test case illustrating the problem.

I tried two approaches to solve the problem but both weren't really straightforward and I'm not sure which one to pursue further.

1. Use @query\_id@ in the hidden fields of the modal instead of serializing all the filters. I changed the @query\_as\_hidden\_field\_tags@ helper for this, basing it's output on @Query#as\_params@. That worked somewhat but at least the 'all columns' option would need special treatment in this case because it was ignored.
2. Change QueriesHelper#retrieve\_query to not initialize the new query with the project when creating it, but apply the params first and set the project later. While this solved my case, it broke a lot of other test cases in both issues and timelog\_reports controller tests.
3. There might be a better way to fix this in @IssueQuery@ itself, by differentiating the cases of 'should I show the project filter in the UI' (only if called outside project context) from 'should I apply an already set project filter' (yes, if the projects filtered for are sub-projects of the present project).

Opinions, anybody?

The problem is present at least since 3.3, I worked on current trunk (r17320).

**履歴**

#1 - 2022/05/10 17:04 - Admin Redmine

- カテゴリ を Issues filter\_56 にセット