

Redmine - Patch #23417

Make the ID column configurable in the issues listing like the other columns

2016-07-25 01:11 - Marius BĂLTEANU

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Issues filter	Estimated time:	0.00 hour
Target version:			
Description			
Currently, in the issues listing table, there are two "hard coded" columns:			
<ol style="list-style-type: none">1. first column with the checkbox2. second column with the issue ID			
In our custom theme, we want to have the subject in the first column in order to improve the readability of the issues and subtasks: id_column.png.			
To obtain that, we hide both columns from the CSS, but we weren't able to display the ID column as last column in the table by default without development (or let the users to decide the ID column position).			
Attached is a patch that:			
<ul style="list-style-type: none">- makes the ID column configurable (like the other columns)- adds the ID column as first column in the default Redmine settings- migrates the default columns settings in order to add the ID column as first column.- updates the existing tests to include this change- removes some empty spaces			

History

#1 - 2016-07-25 17:14 - Anonymous

+1

Very useful patch!

#2 - 2016-07-28 17:50 - Marius BĂLTEANU

- File `make_id_column_configurable_v2.patch` added

The migration file is missing from the first patch. I've attached now the complete patch (`make_id_column_configurable_v2.patch`).

#3 - 2016-08-22 09:52 - Toshi MARUYAMA

- Description updated

#4 - 2016-08-22 09:57 - Toshi MARUYAMA

- "plugins/README" is deleted accidentally.
- db/migrate is too hard to test.
Could you use alternative way?

#5 - 2016-08-22 10:06 - Toshi MARUYAMA

Toshi MARUYAMA wrote:

- db/migrate is too hard to test.
Could you use alternative way?

Oh, I can see `20131004113137_support_for_multiple_commit_keywords.rb` uses similar way.

[source:tags/3.3.0/db/migrate/20131004113137_support_for_multiple_commit_keywords.rb](#)

#6 - 2016-08-22 11:22 - Marius BĂLTEANU

- File `make_id_column_configurable_v3.patch` added

Thanks Toshi for your feedback.

I've updated the patch in order to:

- clearly apply on the current trunk
- remove the accidentally "plugins/README" deletion

For testing the "db/migrate" I used the following scenarios:

- open "settings?tab=issues" and "projects/<project_identifier>/issues?set_filter=1"
- apply the patch without migration
- refresh the admin page and observe that the default columns are still the same (without id)
- refresh the issues list and observe that the ID column is no longer displayed in the issues table
- run the migrations
- refresh the admin page and observe that the ID column now is the first column in the default columns setting
- refresh the issues list and observe that the ID column is displayed now in the issues table.

Regarding the implementation, I don't have another alternative way for the moment. The logic behind the current implementation was:

- get the current default columns setting
- check if the ID column already exists in the settings (on a standard Redmine setup (without patches and plugins) this case cannot exist, but I wanted to cover other cases where a plugin or a patch already made this column configurable)
- if exists, do nothing
- if doesn't exist, add the id column as first column
- save the new settings.

Please let me know if you still want to find another way for migrations or make other changes to the patch.

#7 - 2016-08-22 12:13 - Marius BĂLTEANU

Toshi MARUYAMA wrote:

Toshi MARUYAMA wrote:

- db/migrate is too hard to test.
Could you use alternative way?

Oh, I can see 20131004113137_support_for_multiple_commit_keywords.rb uses similar way.
[source:tags/3.3.0/db/migrate/20131004113137_support_for_multiple_commit_keywords.rb](https://source.rubygems.org/versions/3.3.0/db/migrate/20131004113137_support_for_multiple_commit_keywords.rb)

Yes, my implementation was inspired from that file.

#8 - 2016-09-21 05:21 - Toshi MARUYAMA

- Target version set to 3.4.0

#9 - 2016-11-19 09:14 - Marius BĂLTEANU

- File 3.4.0_make_id_column_configurable_v4.patch added

Updated the patch to cleanly apply on the current trunk.

#10 - 2017-01-16 21:07 - Marius BĂLTEANU

The patch is compatible with the latest trunk. @Jean-Philippe Lang, please let me know if is anything wrong with this patch.

#11 - 2017-01-29 12:12 - Jean-Philippe Lang

- Target version deleted (3.4.0)

Marius, nothing is wrong with the patch but having the issue id in first position is part of the UI. Users should not have to handle this when choosing the information that they want to see on the issue list, and I prefer not to apply this change.

Note that saved queries would also have to be fixed in a migration (id column is missing from all saved queries after applying the patch). If you want to always have it in last position, you could alter Query#columns to append the id instead of inserting it in first position.

#12 - 2017-01-29 16:05 - Marius BĂLTEANU

Thanks for your feedback, Jean-Philippe.

Jean-Philippe Lang wrote:

If you want to always have it in last position, you could alter Query#columns to append the id instead of inserting it in first position.

No, I don't want to have it in last position, I prefer to let each user decide the position of the ID column in the issue list. For example, we have some projects where the ID is not added to the issue list because it is irrelevant.

#14 - 2021-07-08 01:26 - ashraf alzyoud

CAN U CREATE PLUGIN for it or update patch plz
i dont need issue id appear in my report (pdf,cvs)

Files

id_column.png	156 KB	2016-07-24	Marius BĂLTEANU
make_id_column_configurable.patch	20.8 KB	2016-07-24	Marius BĂLTEANU
make_id_column_configurable_v2.patch	21.8 KB	2016-07-28	Marius BĂLTEANU
make_id_column_configurable_v3.patch	21 KB	2016-08-22	Marius BĂLTEANU
3.4.0_make_id_column_configurable_v4.patch	20.5 KB	2016-11-19	Marius BĂLTEANU