

Add tags to issues

2022/05/09 13:27 - Admin Redmine

ステータス:	New	開始日:	2008/06/14
優先度:	通常	期日:	
担当者:		進捗率:	0%
カテゴリ:	Issues_2	予定工数:	0.00時間
対象バージョン:	Candidate for next major release_32	作業時間:	0.00時間
Redmineorg_URL:	https://www.redmine.org/issues/1448	status_id:	1
category_id:	2	tracker_id:	2
version_id:	32	plus1:	70
issue_org_id:	1448	affected_version:	
author_id:	1309	closed_on:	
assigned_to_id:	107353	affected_version_id:	
comments:	139		

説明

Many of our issues span multiple categories -- or more specifically, we'd like to be able to search and manage them with something more flexible than just categories.

It'd be great if there was a way to add tags to issues.

workaround

Use Plugin

https://github.com/ixti/redmine_tags/

journals

Ask - I just added a Custom Field (long text) and made it search-able. Seems to work reasonably well.

It would be better to have global,unified tagging feature so you could select one or more tags to your wiki page, document, forums or issues maybe even changesets.

This kind of tag information could be used to construct more spesific views that can group together all that information.

+1 for tags, is one stopper to migrate our system to redmine

+1

A true tagging system would be very handy.

+1 I think this would be a great feature. There are several tagging plugins for Rails already that we can use.

+1

+1 Would improve usability...

+1 like this idea very much!

+1

Add tagging support for projects as well. Being able to tag a project "opensource" versus

"proprietary" along with other tags would be very useful.

So anyone have any plans to work on this? Otherwise I'll take a crack at it. It looks like some people want tagging all around the system, but I would probably limit it to tickets for the time being since that's all I really need, and leave it up to other people that want other features (besides I've never hacked around with the Redmine source so this will be my first foray into that ;))

+1!

Tags could substitute category. It's very web 2.0! ;)

How This Custom Field with the searchable option will be used as tag.I want this Tagging functionality in News,Issues,Documents,Forums,Wiki,Messages,Changesets

Custom fields can be created only for Projects,Issues and users as far as i Know

For Tagging Documents under Particular tagging is there any Plugin Available

I want to implement This Tagging functionality in all possible places in Redmine.How can i accomplish this functionality.Please guide me in this issue

+1

+1

+1

I do not mind tags, but they will not be able to replace Categories.

Currently categories have default assignee. This means: create issue, drop it to a category, and issue gets auto assigned. With tags you will not be able to resolve who is auto assigned to issue. And assigning multiple people only makes everyone think that 'the other guy will do it'.

Tags yes, but do not remove Category, unless someone has a solution on auto assign an assignee from tags?

This is related to #480 - "Wiki: Support categories tagging and autolisting" and the more general issue #2897 - "Tagging in Redmine".

Personally tagging would be good for the wiki as pages don't always fit into a neat category so being able to add tags is one step better than just relying on the good search facilities in Redmine.

+1

+1

+1

+1

+1

I really miss this feature. For my needs, it could replace category in tickets.

+1 from me too.

+1

+1

The possibility of tagging issues is a minimum to identify areas affected by them. Ok, it is not like the more desirable feature of mapping issues with the configuration items affected but as currently there is no plugin providing an inventory or Configuration Management DataBase I think that tagging is an interesting first step and more than enough for most of us.

I agree that tagging is not a substitute for categories.

+1

+1 (I'll also check if there's a "vote" feature in redmine :))

We don't use redmine yet and I thought that the categories are actually the tags, but further investigation reveals that is not. Sad. It won't take much time to anybody who knows ruby to add one more enum field like categories and add possibility to search issues by that field.

Of course it must be possible to add multiple tags.

+1

+1

<http://github.com/mbleigh/acts-as-tagable-on>

is a very active plugin, and easy to setup ;)

+1

+1

+1 ... this would be a major step towards completely, truly ruling the world.

Tags are nice but I find them to get really disorganized due to users putting in either random or excessive amounts of words, which is why I favor admin/manager controlled categories they can select from.

You could always go the way of GMail and allow issues to be assigned multiple categories, instead of creating an entirely separate feature that does more or less the same thing (in my opinion).

I've just completed a prototype with this for a client using acts-as-tagable-on. Once I get their permission, I'm going to try to extract it into a plugin or core patch.

Mike K: Tags can become disorganized but I find that's better than having to define one Category per issue. I'm thinking about adding some permissions for editing tags, so you can restrict who can re-tag an issue (e.g. UserA can add new tags, UserB can only add existing tags).

Eric Davis wrote:

permissions for editing tags, so you can restrict who can re-tag an issue (e.g. UserA can add new tags, UserB can only add existing tags).

I like that idea and solves my only concern ;)

Letting privileged users manage (add/edit/delete) the tags via some sort of panel would be needed if you plan to replace categories though, especially for newly created projects so you can define new tags they can use without being forced to add new issues in order to do so.

Eric Davis wrote:

I've just completed a prototype with this for a client using acts-as-taggable-on. Once I get their permission, I'm going to try to extract it into a plugin or core patch.

Mike K: Tags can become disorganized but I find that's better than having to define one Category per issue. I'm thinking about adding some permissions for editing tags, so you can restrict who can re-tag an issue (e.g. UserA can add new tags, UserB can only add existing tags).

Eric,

Great to hear! The permission model you laid out seems like it will work well too. Might you also want to have a permission which allows the user to assign or unassign tags to the issue? That way, you can have a user (in my case a customer) who can not edit the issue but can tag it with existing tags. That way, they can affect "state" change without really being able to edit the issue. If you did it this way, it seems like the two permissions are:

- manage tags
- assign/unassign tags to issues

Will you be able to share tags across projects?

Mike K wrote:

Letting privileged users manage (add/edit/delete) the tags via some sort of panel would be needed if you plan to replace categories though, especially for newly created projects so you can define new tags they can use without being forced to add new issues in order to do so.

I don't want to replace categories yet, just add tags as a more flexible option. Maybe later we can merge categories and tags.

Randy Syring wrote:

Great to hear! The permission model you laid out seems like it will work well too. Might you also want to have a permission which allows the user to assign or unassign tags to the issue? That way, you can have a user (in my case a customer) who can not edit the issue but can tag it with existing tags. That way, they can affect "state" change without really being able to edit the issue.

I was thinking:

- view tags
- add new tag - e.g. adding a new tag
- manage tags on issue - e.g. editing tags on an issue, would only use existing tags

Will you be able to share tags across projects?

Yes, I'd think that would be needed.

I know it's a different subject, but I think tags should be employed more widely in redmine. It would be useful for documents as well.

Thanks

Eric Davis wrote:

Maybe later we can merge categories and tags.

IMHO it's better to keep both, categories and tags. Or, for example rename "categories" with "components".

Is there any progress on this issue?

Eric, I would like to offer my help, if you need some hands more :))

+1

Eric, are there any news?

Could this feature make it v1.0.0?

Philippe Lafoucrière wrote:

I know it's a different subject, but I think tags should be employed more widely in redmine. It would be useful for documents as well.
Thanks

I agree,

tags would be great for issues, but they are a must for documents

+1

To clarify tags vs categories:

As patterns in tagging emerge, you can "pave the cow path" by turning some tags into categories (while allowing other tags to remain a cross-referencing and grouping tool).

Tags are a little like people writing in the margins on a well-established paper form. They're a great way to determine how the form can be improved.

Adding tags is an extremely high-leverage activity. It will free up productivity for all Redmine sites and end users.

+1

Tagging a la Trac would support many use cases!

How will tags be protected from inappropriate deletion by users with insufficient rights?

I am stepping down from working on Redmine. If someone else is interesting in working on this issue, feel free to reassign it to them.

Eric Davis

I have some success on implementing tags on base of acts_as_taggable on steroids https://github.com/jviney/acts_as_taggable_on_steroids. Soon I'll push it onto github, so if core team will be interested I'll be very proud to share the patch.

My fork is based on top of Eric's git repo and available as a separate branch here: https://github.com/ixti/redmine/tree/taggable_issues

Right now everything works. The only thing that is left is to make representation more sexy :))

Ok, now everything is ready, the only thing that left is some css styling eye-candies.

What are the different efforts trying to implement this? I see there's also a "@redmine_tagging@":https://github.com/friflaj/redmine_tagging plugin: which one is more mature?

Hmm... seems like I was to hurry in my wish to implement this. :)) Next time will google harder. I believe redmine_tagging is more mature.

Although, probably, redmine_tagging is more mature (I will install it to see what it can and what it can't), I believe that tagging should be core feature (you don't use categorizing issues like a plugin, don't you?) and not pluggable. Anyway I have improved my fork, so now tags are looking more nice (with graceful downgrading), filtering works better and more predictable, settings are more clean. The only thing my fork miss is it's only for issues, but I don't see any problems to extend it to support wiki.

+1

+1

Tagging is a must

I have finished my "redmine_tags":https://github.com/ixti/redmine_tags/ plugin, which gives tagging functionality. It's features:

- Issues can have tags
- Tags are entered into the input text box separated by comas (if no JS support), else it have more or less nice input for the tags
- Sidebar int the issues list can show tags of the project as cloud of tags, or list, or it can be disabled.
- You can switch on or off displaying of related (with tag) issues

- All tags are rendered as links to the issues filter page with predefined filter with chosen tag

+1

I installed redmine_tags and found that the filtering didn't work.

clicking on a tag in the sidebar's tag cloud did not filter the list of issues by tag
- all issues were still displayed and the order did not change either.

there was no 'tags' filter available in the filter drop-down control.

there is no support for tagging wiki pages.

unsure of support for searching for tags using the search bar.

I then tried the redmine_tagging plugin and that appears to work correctly. (that is, my users are not confused by it).

The above was on performed on redmine 1.1.2.

Dominic,

Which version of redmine_tags you were using? Because the problem you are claiming about was solved long time ago :)) Current version have all the features I wanted. The only thing it still lack of is wiki tagging. But there's no easy way to add tags to wiki because of internal structure of wiki in redmine. I'm going to try to solve this with some hacks, but right now I simply have no time to spent it on redmine_tags :))

Sorry, that my work confused your clients. And that's great that redmine_tagging finally solved all your problems. :))

I definitely advocate for replacing categories with tags instead of adding tags. There's no need to have duplicated functionality, which will just confuse users.

Alexey Skor wrote:

I definitely advocate for replacing categories with tags instead of adding tags. There's no need to have duplicated functionality, which will just confuse users.

It's not that simple. See e.g. notes "17":<http://www.redmine.org/issues/1448#note-17>, "26":
<http://www.redmine.org/issues/1448#note-26>, "37":<http://www.redmine.org/issues/1448#note-37> and "43":
<http://www.redmine.org/issues/1448#note-43>. The issue categories are also used in other parts of Redmine like e.g. the versions view and the issues report. So imho it is questionable if these two are actually providing duplicate functionality.

I personally like the "solution" proposed by Aleksey Zapparov in note "43":
<http://www.redmine.org/issues/1448#note-43>.

+1 Tagging would be incredibly useful within our Wiki.

The "redmine tags plugin" works fine for that requests.

Terence Mill wrote:

The "redmine tags plugin" works fine for that requests.

I cannot find "redmine tags plugin".
Do you mean "Tagging plugin"?

http://www.redmine.org/projects/redmine/wiki/Plugin_List#Tagging-plugin

That list of plugins is little bit outdated. The new one is <http://www.redmine.org/plugins>

I believe Terence Mill was talking about: https://github.com/ixti/redmine_tags (on the plugins list it has little bit old version).

Yes, exactly this plugin i meant.

+1 Tagging would be incredibly useful

+1 I see native tagging as one of the only killer features left in Redmine. The power of folksonomy is quite staggering in systems like these.

+1

+1

+1

+1

Global tags help linking stuff together.

+1

Agreed. Currently I can't do categorize an issue as both SEO and performance, which is a bummer.

+2 !:)

Really helps when searching doesn't really work (e.g. when people use different terms to describe the same problem).

+1 for me too

Another +1 .. I'll be testing out plugins for now, as a single category for issues is not sufficient for our needs - but unfortunately none of the tagging plugins seem particularly 'maintained', and this seems like a very valuable addition to the core. Is there any sort of plan or target version for this? I saw reference to tagging support planned for 1.3 "here": <http://www.redmine.org/news/48> - I'm just not sure if there's any more recent information.

+1

Another +1 (a voting plugin on these issues would probably help reduce the number of +1 posts :P ... maybe #6945 ...)

Tagging functionality being rolled directly into Redmine would be beneficial for more two (2) reasons:

- 1) It's extremely useful and would make the core application a lot more flexible
- 2) It would allow plugin developers a unified way of offering this functionality in their extensions

The second point I think may be more important, because as it stands there are a number of plugins that "reinvent the wheel" in order to offer tagging support. This results in collisions, migration failures and other errors if two or more plugins that require tagging functionality are installed.

If this would be seriously considered, I'd also propose making tagging a function that can be extended to documents, wiki pages, news items ... etc. I'm sure there would need to be some considerations regarding permissions for visibility and filtration so I don't assume this is an easy solution ;)

+1

+1

+1

+1

+1

However there is a plugin which works on redmine 2.1

https://github.com/ixti/redmine_tags.git

Assigned issue with no assignee back to New status.

+1

this could also be a reasonable solution for #9306 and #1767

+1

+1

+1

+1

this would be very usefull, and much mor flexible than categories or hierarchy. please, add it to the roadmap !

+1 from me too.

+1

+1

+1

+1

Used tags (or Custom Fields) with Bugzilla for years. And, honestly - this is one and only feature I had in Bugzilla and miss in Redmine.

We use https://github.com/ixti/redmine_tags/ and have been very happy with the feature minus a few hicups.

+1

Yes, +1 to add this to the roadmap. It would be very useful, and much more flexible than categories.

+1 - is it possible to collate all the tags that have been added

+1 or +2 :-)

this one looks also awesome:

!{width:600px}<http://redminecrm.com/attachments/download/3654/contacts.png>!

+1

+1

I would love to use such a feature. I mean, many-to-many relation like tags, not a many-to-one relation like categories and versions in redmine. Often I have issues that need to be in more than one group and once you selected category or version you cannot include issue in other group. I would like something more flexible that I can use later for filtering and querying.

I have found these two plugins for this:

https://github.com/ixti/redmine_tags/

and

https://github.com/Undev/redmine_tagging

Not sure witch one is best, but I do not dare to use them as I use redmine ubuntu package for development team and plugin installation in redmine requires rake commands. It does not seem easy and unrisky. I am afraid I may break something in my redmine installation and I have already had integrity db problems with redmine in past. I would love plugin installation was in redmine as easy as other products (jenkins...)

I would love to use this feature.

https://github.com/ixti/redmine_tags works with latest Redmine versions.

If the Redmine contributors are interested to add the tags functionally to issues and wiki pages, we can provide a patch that integrates the redmine_tags plugin into Redmine core (I'm one of the plugin maintainers).

Marius BALTEANU, https://github.com/Undev/redmine_tagging also works with latest redmine.

and yes,

i would be great to support tags "out of the box".

+1

+1

+1 - it'd be great to have tags available in the default install without having to install plugins

+1

+1

+1

I would like to use multiple tags, including subprojects.
I want to set the color of each tag.

+1

Out-of-box tags will be a killing feature!

I've started work on adding the tag functionality to Redmine based on the plugin https://github.com/ixti/redmine_tags (to which I've contributed in the last 2 years).

My plan is to add in a first phase a simple tagging feature to issues and after the functionality is approved/added to the core, to extend it to other entities (e.g. wiki, projects), propose all other features (e.g.: bulk edit tags, colored tags) offered by the mentioned plugin and more.

This first phase consists of two patches:

1. Add tags to issues (the core functionality)
2. add/remove tags
3. show tags on issue page
4. filter issues after tags
5. tags as column in issues list
6. export tags (pdf, csv)

This patch is ready and can be tested using the attached patch. I'm wondering if we should have a permission to manage tags (from my point of view, the edit issue permission is enough) or if something else is missing and it should be in this first phase.

1. Add a JS library to handle the tags in the UI (including autocomplete) In the attached patch I propose "Selectize.js": <https://selectize.github.io/selectize.js> because I know that Jean-Philippe Lang prefers to not add Select2 to the core and tag-it.js library (used by the plugin) is no longer maintained. The patch is not ready, but I would like to confirm first that I can use this library (other proposals are welcome) and if it should be configurable from which projects should the autocomplete propose the tags (From all projects, From project tree, From project hierarchy, From subprojects).

This feature is one of the most awaited features, it is ranked as no 1 in the

message#51899, it has more than 120 watchers (80 from this issue and 45 from #2897), multiple comments and related issues. Also, plugins can benefit if the tagging feature it'll be part of the core. I would love to work on this and deliver it in the next version after Redmine version:4.0.0 (which is almost ready).

This patch is a feature the user has been longing for.

Thank you, Marius BALTEANU!

I've successfully applied the patch.

Editing

!tags-edit.png!

Filtering

!tags-filter.png!

As issue attribute

!tags-show.png!

I agree with adding tagging functionality to the Redmine core.

However, I am worried whether it will conflict with knowledgebase-plugin which uses the same tag function.

Using redmine_tags-Plugin and Knowledgebase-Plugin at the same time fails with DB migration.

I would like to use the tag function of Redmine-core and Knowledgebase-Plugin at the same time.

https://github.com/alexbevi/redmine_knowledgebase/issues/320

```
= 20130513154700 AddTagsAndTaggings: migrating =====  
rake aborted!
```

```
StandardError: An error has occurred, all later migrations canceled:
```

```
A plugin is already using the "tags" or "taggings" tables, and  
the structure of the table does not match the structure expected  
by redmine_knowledgebase.
```

Yuuki NARA wrote:

Using redmine_tags-Plugin and Knowledgebase-Plugin at the same time fails with DB migration.
I would like to use the tag function of Redmine-core and Knowledgebase-Plugin at the same time.

https://github.com/alexbevi/redmine_knowledgebase/issues/320

I think such conflicts should be addressed by plugins. There are numerous plugins in the world. It is impossible for us to take them into account.

Go MAEDA wrote:

Yuuki NARA wrote:

Using redmine_tags-Plugin and Knowledgebase-Plugin at the same time fails with DB migration.
I would like to use the tag function of Redmine-core and Knowledgebase-Plugin at the same time.

https://github.com/alexbevi/redmine_knowledgebase/issues/320

I think such conflicts should be addressed by plugins. There are numerous plugins in the world. It is impossible for us to take

them into account.

I agree. What we can do as extra step is to ensure that the migrations work if you had the redmine_tags plugin installed. I'll check this scenario this weekend.

Go MAEDA,Marius BALTEANU ,thanks for your comments.

I agree with the following two points.

It is necessary for plugin to fix plugin due to Redmine version upgrade.

It is beneficial to Redmine users to integrate redmine_tags in the Redmine core.

However, many people use Redmine_knowledgebase plugin.

Redmine_knowledgebase has twice as many users as redmine_tags users. (Watchers, stars)

It is not a small number,and it has been used for a long period of 8 years.

https://github.com/alexbevi/redmine_knowledgebase 69 watchers,365 stars,163 forks,since 2010

https://github.com/ixti/redmine_tags 27 watchers,158 stars,107 forks,since 2010

I hope that Redmine-core developers, Redmine-plugin developers and Redmine-users will be able to cooperate and efficiently perform the necessary work.

Yuuki NARA wrote:

Go MAEDA,Marius BALTEANU ,thanks for your comments.

I agree with the following two points.

It is necessary for plugin to fix plugin due to Redmine version upgrade.

It is beneficial to Redmine users to integrate redmine_tags in the Redmine core.

However, many people use Redmine_knowledgebase plugin.

Redmine_knowledgebase has twice as many users as redmine_tags users. (Watchers, stars)

It is not a small number,and it has been used for a long period of 8 years.

https://github.com/alexbevi/redmine_knowledgebase 69 watchers,365 stars,163 forks,since 2010

https://github.com/ixti/redmine_tags 27 watchers,158 stars,107 forks,since 2010

I hope that Redmine-core developers, Redmine-plugin developers and Redmine-users will be able to cooperate and efficiently perform the necessary work.

As the original author of the knowledgebase plugin I'm wholeheartedly in support of this (as per the comment at #1448#note-82 :P).

The lack of internal tagging has been a bane on not just my plugin, but many other plugins (including I believe the Redmine CRM plugin, which doesn't have public usage details). I have not been active in the Redmine community for a number of years, but would be happy to help with developing a migration plan for the built in tagging functionality.

I agree the responsibility should ultimately fall on the plugin authors, but a singular migration strategy for anyone that's based their features on @acts-as-tagable-on@ may help get existing plugins updated faster.

Indeed, the migrations from my first patch (which are the default ones generated by acts-as-tagable-on gem) do not work if you have the @redmine_tags@ plugin already installed.

I'm attaching a new patch that merges the migrations in less files and adds also checks on create_table/add_index actions in order to fix the migrations.

I made the following tests and everything looks good:

1. Current trunk with @redmine_tags@ installed + all three patches: rake db:migrate ran successfully.
2. Compared the tables generated by

current trunk + first two patches:

{{collapse(View tables)}}

```
mysql> show create table tags /G;
```

```
***** 1. row *****
```

Table: tags

```
Create Table: CREATE TABLE `tags` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(255) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT NULL,  
  `taggings_count` int(11) DEFAULT '0',  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `index_tags_on_name` (`name`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8  
1 row in set (0.00 sec)
```

```
mysql> show create table taggings /G;
```

```
***** 1. row *****
```

Table: taggings

```
Create Table: CREATE TABLE `taggings` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `tag_id` int(11) DEFAULT NULL,  
  `taggable_type` varchar(255) DEFAULT NULL,  
  `taggable_id` int(11) DEFAULT NULL,  
  `tagger_type` varchar(255) DEFAULT NULL,  
  `tagger_id` int(11) DEFAULT NULL,  
  `context` varchar(128) DEFAULT NULL,  
  `created_at` datetime DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `taggings_idx` (`tag_id`,`taggable_id`,`taggable_type`,`context`,`tagger_id`,`tagger_type`),  
  KEY `index_taggings_on_taggable_id_and_taggable_type_and_context` (`taggable_id`,`taggable_type`,`context`),  
  KEY `index_taggings_on_tag_id` (`tag_id`),  
  KEY `index_taggings_on_taggable_id` (`taggable_id`),  
  KEY `index_taggings_on_taggable_type` (`taggable_type`),  
  KEY `index_taggings_on_tagger_id` (`tagger_id`),  
  KEY `index_taggings_on_context` (`context`),  
  KEY `index_taggings_on_tagger_id_and_tagger_type` (`tagger_id`,`tagger_type`),  
  KEY `taggings_idy` (`taggable_id`,`taggable_type`,`tagger_id`,`context`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

}}

current trunk + all three patches:

{{collapse(View tables)}}

```
mysql> show create table tags /G;
```

```
***** 1. row *****
```

Table: tags

```
Create Table: CREATE TABLE `tags` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(255) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT NULL,  
  `taggings_count` int(11) DEFAULT '0',  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `index_tags_on_name` (`name`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

```
mysql> show create table taggings /G;
```

```
***** 1. row *****
```

Table: taggings

```
Create Table: CREATE TABLE `taggings` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `tag_id` int(11) DEFAULT NULL,  
  `taggable_type` varchar(255) DEFAULT NULL,  
  `taggable_id` int(11) DEFAULT NULL,  
  `tagger_type` varchar(255) DEFAULT NULL,
```

```
`tagger_id` int(11) DEFAULT NULL,
`context` varchar(128) DEFAULT NULL,
`created_at` datetime DEFAULT NULL,
PRIMARY KEY (`id`),
UNIQUE KEY `taggings_idx` (`tag_id`,`taggable_id`,`taggable_type`,`context`,`tagger_id`,`tagger_type`),
KEY `index_taggings_on_taggable_id_and_taggable_type_and_context` (`taggable_id`,`taggable_type`,`context`),
KEY `index_taggings_on_tag_id` (`tag_id`),
KEY `index_taggings_on_taggable_id` (`taggable_id`),
KEY `index_taggings_on_taggable_type` (`taggable_type`),
KEY `index_taggings_on_tagger_id` (`tagger_id`),
KEY `index_taggings_on_context` (`context`),
KEY `index_taggings_on_tagger_id_and_tagger_type` (`tagger_id`,`tagger_type`),
KEY `taggings_idy` (`taggable_id`,`taggable_type`,`tagger_id`,`context`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
}}
```

and are the same.

Also, I've set the migrations version to 4.2 in order to preserve the type of id column to int instead of big int (default behaviour in Rails 5). Maybe we will handle this change to all Redmine tables in another ticket.

@Alex Bevilacqua, should I open a ticket on GitHub to test the compatibility between these patches and your plugin? Also, if you have a better idea about how we should handle these migrations, please let me know.

Marius BALTEANU wrote:

@Alex Bevilacqua, should I open a ticket on GitHub to test the compatibility between these patches and your plugin? Also, if you have a better idea about how we should handle these migrations, please let me know.

I've linked your comment to an existing issue (https://github.com/alexbevi/redmine_knowledgebase/issues/320). I'll see if I can slot in some time to get this tested. Even though I'm no longer an active maintainer, I'd like to see this resolved as it really has been a pain in my ass for the better part of a decade :P

Alex Bevilacqua wrote:

I've linked your comment to an existing issue (https://github.com/alexbevi/redmine_knowledgebase/issues/320). I'll see if I can slot in some time to get this tested. Even though I'm no longer an active maintainer, I'd like to see this resolved as it really has been a pain in my ass for the better part of a decade :P

Thanks, I've subscribed to the issue. Hopefully we'll see this integrated soon.

I've linked your comment to an existing issue (https://github.com/alexbevi/redmine_knowledgebase/issues/320).

Both tickets seems to wait for each other now. Any news about this nice feature?

Dan Untenzu wrote:

I've linked your comment to an existing issue (https://github.com/alexbevi/redmine_knowledgebase/issues/320).

Both tickets seems to wait for each other now. Any news about this nice feature?

Yes... i'm waiting for this great feature to become a core feature of Redmine.

Updated the patch to apply cleanly on the current trunk.

I would like to discuss this for the next major release, it is one of the most requested feature.

It would be great to have tags as a core feature of Redmine.

Can't wait for this feature in core!

There is a free version of commercial plugin: <https://www.redmineup.com/pages/plugins/tags>

But it is so far from being good... For example there is no way to setup per-project tags. There is now way to enable tags usage for just some project. Sometimes it breaks "edit issue" form. And so on.

Any chance of adding this functionality?

Is there a patch set which applies against 4.1.1? A simplistic attempt had some hunks fail to apply.

Thanks.

In case it is useful for anyone; after waiting a lot of years for this in mainstream: I achieved this with Custom field key/value list in a way is quite satisfactory for me.

- It can be defined globally and used in more than once project (categories for instance are defined locally to project)
- Being key/value list "labels/tags" can be renamed after creation; other custom fields require a lot of work on issues for renaming. You can use any text knowing that you can change it easely.
- Each issue can have more than one "label/tag" as expected. This is, M:N or many-to-many relations, I think the biggest problem that solves tags/labels, and in redmine no other field has this with the exception of issue-to-issue relations, that is not what we want in many use cases, so this is why tags/labels are missed so much in redmine.
- It is in mainstream branch so available on any recent redmine installation.
- You can repeat this several times, having more than one group of labels.
- You can filter issues easily by active tag.

What you can't do anyway:

- Use this labels for sorting. (minor issue, it would be nice and it is easy to implement, now yo can use an extra numeric custom field for this).
- Labels must be defined by administrator by it is not a problem for my use cases.

+1

FWIW -- the following plugin is a nice solution while we wait for native tags.

https://github.com/AlphaNodes/additional_tags

related_issues

relates,New,2897,Tagging in Redmine

relates,New,29663,Select a JavaScript library to handle tags

duplicates,Closed,9616,replace "issue categories" with labels

履歴

#1 - 2022/05/10 17:28 - Admin Redmine

- カテゴリ を Issues_2 にセット

- 対象バージョン を Candidate for next major release_32 にセット

#2 - 2022/05/11 13:56 - Admin Redmine

- workaround を更新

- journals を更新
- related_issues を更新