Redmine - Feature #36162

Add notification reason to the email instead of the default static email footer

2021-11-08 23:09 - Marius BĂLTEANU

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Marius BĂLTEANU	**% Done:**	0%
Category:	Email notifications	**Estimated time:**	0.00 hour
Target version:	6.0.0		
Resolution:			

Description

The default email footer setting is:

> You have received this notification because you have either subscribed to it, or are involved in it.
> To change your notification preferences, please click here: http://hostname/my/account

This setting can be changed from Settings -> Notifications -> Email footer.

This approach have several issues:

- the text can be removed by an admin
- the text is not translatable
- the text is too generic and the user cannot understand why he received that notification

My proposal is to replace this text with the real reason of why the user received the notification. In this phase, I propose the following reasons with the according translations:

Reason	Message	Details
INVOLVED	You have received this notification because you are involved in.	user is either author, assignee or previous assignee
MENTIONED	You have received this notification because you have been mentioned in.	user in mentioned in that event
SUBSCRIBED	You have received this notification because you have subscribed to it.	user is subscribed to that event
WATCHER	You have received this notification because you are watching it.	user is a watcher
ADMIN	You have received this security notification because you are an administrator.	user is an admin and he received a security notification

In the future, we can add even a more granular reason for the involved reason: author, assignee or previous assignee.

Also, I propose to always display the text "To change your notification preferences, please click here: %{link to my account}."

Below is an example from a security notification with both messages (first message is the new one; second message is the old one).
sn.png

Related issues:

Related to Redmine - Feature #13919: Mention user on issues and wiki pages us...	**Closed**
Related to Redmine - Feature #38492: Provide some ways to find the issues whe...	**New**

History

#1 - 2021-11-08 23:10 - Marius BĂLTEANU

- Description updated

#2 - 2021-11-08 23:15 - Marius BĂLTEANU

- Related to Feature #13919: Mention user on issues and wiki pages using @user with autocomplete added

#3 - 2021-11-09 07:48 - Bernhard Rohloff

To me, this sounds like a good improvement of user experience.
It also helps to set the personal notification settings appropriately.
+1 from my side.

#4 - 2021-11-09 08:35 - pasquale [:dedalus]

- File bugzilla-sample.png added

+1 for me.
I suggest a more coincise way (like bugzilla): if anyone has more reasons to receive an email, we can summarize with a bullet list as shown in attached screenshot:

> **You are receiving this mail because:**
>
> - You are on the CC list for the bug.
>
> Configure your email settings at https://bugzilla.mozilla.org/userprefs.cgi?tab=email.

#5 - 2021-11-09 08:38 - C S

+1 because it is a really good idea!

#6 - 2021-11-09 11:08 - Bernhard Rohloff

pasquale [:dedalus] wrote:

> ... we can summarize with a bullet list as shown in attached screenshot:

Nice input. I like it.

#7 - 2022-03-29 07:03 - Marius BĂLTEANU

- Target version changed from Candidate for next major release to 5.1.0

#8 - 2022-04-26 15:20 - Miodrag Milic

Why not adding MENTIONED reason now?

#9 - 2023-05-04 02:10 - Go MAEDA

- Related to Feature #38492: Provide some ways to find the issues where an user Is mentioned added

#10 - 2023-10-02 23:57 - Go MAEDA

- Target version changed from 5.1.0 to 6.0.0

#11 - 2024-01-17 06:13 - Marius BĂLTEANU

- File 0001-Add-notification-reason.patch added

- Description updated

I've added the patch that I would like to commit in the following days.

The patch introduces a new class that keeps the UserNotificationReasons and two methods that:

- return the reason priority as integer
- reorder an array of users based on the notifications reason.

In mailer.rb, instead of uniquely append users, we add all the users (included duplicates), reorder them based on priority reason and then remove duplicates. In the end, user still receives only one notification, but the one with the highest reason priority. So, if a user is subscribed to all events from a project and also is assigned to issue A, when issue A is updated, he received the "INVOLVED" and not the "SUBSCRIBED" reason.

Also:

- the text reason from email is translatable now, so each user will receive the message based on his language

- the reason is available also as email header

#12 - 2024-01-17 06:19 - Marius BĂLTEANU

I'm not sure about two things:

1. Add all reasons to the email as Pasquale suggested in #note-4.
2. Add a migration that removes from the database the default value for emails_footer if this value was not changed (except the generic link to /my/account).

Any feedback is really appreciated, most of the work is done so we can include this in the next major release.

#13 - 2024-01-17 06:19 - Marius BĂLTEANU

Miodrag Milic wrote in #note-8:

> Why not adding MENTIONED reason now?

The patch already includes this reason.

#14 - 2024-01-17 07:52 - Marius BĂLTEANU

- Description updated

#15 - 2024-01-18 01:59 - Go MAEDA

How about showing the notification reason in the header instead of the footer? It would be useful to know the reason before you start reading the email to decide if you need to read the entire email.

For example, on some projects I only want to read emails that are assigned to me and don't want to read emails that someone added me as a watcher.

#16 - 2024-01-18 02:33 - Marius BĂLTEANU

Thanks for your feedback!

Can you give me an example of such type of mail with reason in the header?

#17 - 2024-01-18 15:24 - Holger Just

Some random remarks regarding the patch in #note-11:

- I'm not a huge fan of storing transient information on a model instance with an attr_accessor (here the notification_reason). This makes it rather hard to reason about what is actual (stored) model data and what is transient data. As this is only used to transport data between "unrelated" code areas, this also causes spooky magic at a distance. If possible, we should send the data directly. For example, we could define a new value class (or decorator or wrapper) which contains a reference to the user and create objects of those when collecting the notified users. We could then pass these new objects to the mailer, rather than just plain users.
  - This could well be the UserNotificationReason class (possibly with a slightly different name then)
  - Maybe this isn't even needed as it appears that we can always (?) detect the notification reason from the method called in the Mailer.deliver_* methods alone. As each called method called to collect notified users by the deliver_* methods returns users for only one notification reason, we could localize this knowledge in the Mailer.deliver_* methods
  - If this is not enough, we could also add new helper methods to the Issue, WikiPage, News, ... classes which wrap the output of the existing methods into the new wrapper class.
- The UserNotificationReason class should not live in app/models, as it is not a model. It should be in lib/redmine instead.
- You are using map instead of each multiple times (e.g. in the app/models/issue.rb patch). As map is generally used for its returned value rather than the side-effects of the block, this is confusing.
- There could be multiple reasons why a user receives an email, such as (1) the mail is high priority and (2) the user is the author and (3) the user is a watcher. We should be able to express these multiple concurrent reasons in the mail headers and the rendered bodies.

What do you think?

#18 - 2024-01-22 13:12 - Marius BĂLTEANU

- File all_reasons.png added

Holger Just wrote in #note-17:

> Some random remarks regarding the patch in #note-11:

Thank you for taking your time to reviewing the patch! I had concerns about the chosen solution, but I started to work at this for so long time that I just wanted to have a first working version.

- I'm not a huge fan of storing transient information on a model instance with an attr_accessor (here the notification_reason). This makes it

rather hard to reason about what is actual (stored) model data and what is transient data. As this is only used to transport data between "unrelated" code areas, this also causes spooky magic at a distance. If possible, we should send the data directly. For example, we could define a new value class (or decorator or wrapper) which contains a reference to the user and create objects of those when collecting the notified users. We could then pass these new objects to the mailer, rather than just plain users.
    - This could well be the UserNotificationReason class (possibly with a slightly different name then)

Something like this?

UserNotificationRecipient as a value object class with two attributes:

- user: stores the user
- reason stores the reason

All notified_users methods return UserNotificationRecipient objects instead of User. If we choose this option, we can move more logic related to notifications from User class to UserNotificationRecipient class.

Then, in the Mailer class we can use some kind of service to build the notification recipients by iterating through all the UserNotificationRecipient objects in order to concatenate all the reasons or reorder based on priority reason and remove duplicates.

- Maybe this isn't even needed as it appears that we can always (?) detect the notification reason from the method called in the Mailer.deliver_* methods alone. As each called method called to collect notified users by the deliver_* methods returns users for only one notification reason, we could localize this knowledge in the Mailer.deliver_* methods

I think that it's not quite easy because some notified_users methods return a mix of reasons. For example, issue.notified_users can return involved users (author, assignee, previous_assignee), users subscripted to project events (project.notified_users) and users notified about high priority issues. If project.notified_users can be extracted, I think the "involved" and "high priority" require more changes. If we want to do this, we can extract each reason in its own method and concatenate all of them in deliver_issue_ methods.

- If this is not enough, we could also add new helper methods to the Issue, WikiPage, News, ... classes which wrap the output of the existing methods into the new wrapper class.

- The UserNotificationReason class should not live in app/models, as it is not a model. It should be in lib/redmine instead.

I'll take this into consideration.

- You are using map instead of each multiple times (e.g. in the app/models/issue.rb patch). As map is generally used for its returned value rather than the side-effects of the block, this is confusing.

I'll review this!

- There could be multiple reasons why a user receives an email, such as (1) the mail is high priority and (2) the user is the author and (3) the user is a watcher. We should be able to express these multiple concurrent reasons in the mail headers and the rendered bodies.

Just to be double check, you're in favour of displaying all the reasons as Pasquale suggested in #note-4, something like that:

all_reasons.png
What do you think?

To summarise this, I'm fully open to rework this path and I'm waiting for your feedback before starting again to work on this.

#19 - 2024-02-16 11:06 - Dennis Buehring

Hi,

i wanted to add my two cents to this, because we just had to switch from a redmine_mentions plugin to the builtin functionality, because both where showing the list of users on top of each other.
The plugin sent extra notifications that were easy to spot and treat differently in my mailbox (and all my colleagues), the builtin mentions just sends a regular notification email... everybody just assumed the mentions did not work anymore...

Mentions are a special case, which should be treated differently. I want to inform someone, regardless of his default mail notification options..
If this depends on their settings, why would anyone use this feature ?
If these notifications get lost between the other hundreds of notifications, why would anyone use this feature ?

The plugins did it right, why reinvent the wheel (as a square) after ignoring it for several years ?? :)
This feature is shit in its current form, and one should at least be able to disable it to use the plugins again ;)

#20 - 2024-03-20 00:18 - Marius BĂLTEANU

- File 0001-WIP.patch added

Holger Just, I made a patch (WIP - only for demo purposes) with an alternative implementation using Service Objects. I know that we are not currently using this kind of pattern, but I think it will be useful to start using it in order to cleanup a little bit the models. Also, the patch addresses only the "design" issue caused by attr_accessor.

The patch contains the following:

- The UserNotificationRecipient that stores the User and the reason
- A BuildService that have static methods for each Mailer.deliver_* method and that return all the recipients
- A Builder::Base that is extended by builders for each object, for example Builder::Issue or Builder::Journal
- All Builder::* methods return UserNotificationRecipient objects.

In theory, if we move forward with this implementation, we can get rid of all the methods related to notifications from models. Beside this new pattern, another disadvantage is the change that could be quite big.

What do you think? Also, if you have in mind another solution that require less changes, please let me know (if you can show your ideea with a quick patch, it will be great).

#21 - 2024-03-25 07:13 - Jens Krämer

I'd like to add a general +1 for moving things out of the ActiveRecord models :)

Few remarks about the patch:

- I believe you could replace all uses of protected with private and everything should still work?
- I'd like to suggest moving the UserNotificationRecipient inside the namespace of the service. It's a class that most certainly won't be used outside of this context so really does not have to live in app/models. How about NotificationRecipients::NotifiedUser?
- the static methods in BuildService might be moved up one level into the `NotificationRecipients` module

#22 - 2024-03-25 22:08 - Marius BĂLTEANU

Jens, thanks for your feedback, I tried to incorporate your changes in the attached patch (0001-WIP_v2.patch), but I'm not sure about last point because NotificationRecipients ::BuildService was already in the namespace. Please correct me if I'm wrong:

- notification_recipients -> NotificationRecipients
  - notified_user.rb -> NotificationRecipients::NotifiedUser
  - build_service.rb -> NotificationRecipients::BuildService
  - builder
    - base.rb -> NotificationRecipients::Builder::Base
    - issue.rb -> NotificationRecipients::Builder::Issue
    - journal.rb -> NotificationRecipients::Builder::Journal

Once we agree on the structure, I will continue working on the patch.

#23 - 2024-03-25 22:10 - Marius BĂLTEANU

- File 0001-WIP_v2.patch added

#24 - 2024-04-11 12:08 - Holger Just

Thanks Marius for your efforts! Sorry for the late reply, I'll try to check your patch and provide feedback soon.

For now, as a further extension, as requested / mentioned by Patrick Donnelly on IRC, we may want to include the notification reason(s) in a mail header too to allow easy filtering.

#25 - 2024-04-12 07:54 - Marius BĂLTEANU

Holger Just wrote in #note-24:

> For now, as a further extension, as requested / mentioned by Patrick Donnelly on IRC, we may want to include the notification reason(s) in a mail header too to allow easy filtering.

It's already taken care in the patch, it just need to be adapted based on the decision to show only one reason (high highest priority) or all reasons.

Files

sn.png	109 KB	2021-11-08	Marius BĂLTEANU
bugzilla-sample.png	21.6 KB	2021-11-09	pasquale [:dedalus]
0001-Add-notification-reason.patch	35.5 KB	2024-01-17	Marius BĂLTEANU
all_reasons.png	166 KB	2024-01-22	Marius BĂLTEANU
0001-WIP.patch	10.2 KB	2024-03-19	Marius BĂLTEANU
0001-WIP_v2.patch	10.2 KB	2024-03-25	Marius BĂLTEANU