

Redmine - Defect #40610

Projects page never finishes loading

2024-04-22 10:29 - James H

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Performance	Estimated time:	0.00 hour
Target version:	5.0.9	Affected version:	4.1.0
Resolution:			
Description			
Hello,			
I am in the process of upgrading to 5.x.x from 4.x.x and noticed that there was a change to the projects page that is causing me problems, probably due to my open project count.			
I have a lot of open projects at a given time and the changes are causing it to lag out and freeze before it finishes loading...			
I managed to make it a bit better by removing all the project descriptions and filtering from the page but it also still freezes near the end.			
Any way to make this more performant?			
Related issues:			
Related to Redmine - Feature #29482: Query system for Projects page		Closed	

History

#1 - 2024-04-23 08:25 - Go MAEDA

Does the following change improve performance?

```
diff --git a/app/helpers/application_helper.rb b/app/helpers/application_helper.rb
index 910f1f431..24160641b 100644
--- a/app/helpers/application_helper.rb
+++ b/app/helpers/application_helper.rb
@@ -941,7 +941,7 @@ module ApplicationHelper
   if obj.is_a?(Journal)
     attachments += obj.journalized.attachments if obj.journalized.respond_to?(:attachments)
   else
-   attachments += obj.attachments if obj.respond_to?(:attachments)
+   attachments += obj.attachments if !obj.is_a?(Project) && obj.respond_to?(:attachments)
   end
   if attachments.present?
     text.gsub!(/src="( [\^\/"] )+\.(bmp|gif|jpg|jpe|jpeg|png|webp) "\s+alt="( [\^"] *)"?/i) do |m|
```

#2 - 2024-04-23 20:31 - James H

Go MAEDA wrote in [#note-1](#):

Does the following change improve performance?

[...]

yes, that seems to make it load much much faster, taking about 10% of the time it used to take to freeze
Thank you!

#3 - 2024-04-24 05:16 - Go MAEDA

James H wrote in [#note-2](#):

Go MAEDA wrote in [#note-1](#):

Does the following change improve performance?

[...]

yes, that seems to make it load much much faster, taking about 10% of the time it used to take to freeze

Thank you!

Thanks to your feedback, I have identified the cause of the issue you mentioned. I will look into ways to address this issue.

#4 - 2024-04-25 02:26 - Go MAEDA

- File 40610.patch added

- Target version set to Candidate for next minor release

I wrote a patch to fix this issue.

The cause of the issue is that the projects list with the list style loads all attachments whose container_type is Project for each project to render the project's description. Attachments with container_type of Project are files uploaded to the "Files" module.

```
Attachment Load (0.1ms) SELECT "attachments".* FROM "attachments" WHERE "attachments"."container_id" = ? AND "attachments"."container_type" = ? ORDER BY attachments.created_on ASC, attachments.id ASC [{"container_id" : 1}, {"container_type", "Project"}]
↳ app/helpers/application_helper.rb:944:in `+'
Attachment Load (0.2ms) SELECT "attachments".* FROM "attachments" WHERE "attachments"."container_id" = ? AND "attachments"."container_type" = ? ORDER BY attachments.created_on ASC, attachments.id ASC [{"container_id" : 5}, {"container_type", "Project"}]
↳ app/helpers/application_helper.rb:944:in `+'
Attachment Load (0.1ms) SELECT "attachments".* FROM "attachments" WHERE "attachments"."container_id" = ? AND "attachments"."container_type" = ? ORDER BY attachments.created_on ASC, attachments.id ASC [{"container_id" : 6}, {"container_type", "Project"}]
↳ app/helpers/application_helper.rb:944:in `+'
Attachment Load (0.1ms) SELECT "attachments".* FROM "attachments" WHERE "attachments"."container_id" = ? AND "attachments"."container_type" = ? ORDER BY attachments.created_on ASC, attachments.id ASC [{"container_id" : 3}, {"container_type", "Project"}]
↳ app/helpers/application_helper.rb:944:in `+'
Attachment Load (0.1ms) SELECT "attachments".* FROM "attachments" WHERE "attachments"."container_id" = ? AND "attachments"."container_type" = ? ORDER BY attachments.created_on ASC, attachments.id ASC [{"container_id" : 4}, {"container_type", "Project"}]
↳ app/helpers/application_helper.rb:944:in `+'
Attachment Load (0.2ms) SELECT "attachments".* FROM "attachments" WHERE "attachments"."container_id" = ? AND "attachments"."container_type" = ? ORDER BY attachments.created_on ASC, attachments.id ASC [{"container_id" : 2}, {"container_type", "Project"}]
↳ app/helpers/application_helper.rb:944:in `+'
Rendered projects/_list.html.erb (Duration: 188.8ms | Allocations: 68634)
```

The reason attachments are loaded to render text content is to generate HTML tags for inline images. Currently, however, attachments are always loaded even for content that does not contain inline images. This causes the performance issue when displaying many projects in the project list.

The attached patch resolves this performance issue by changing ApplicationHelper#parse_inline_attachments to load attachments only when the text being rendered contains img tags. I guess the change will improve performance not only for displaying the projects list, but also for displaying an issue with large numbers of attachments and comments.

#5 - 2024-04-25 02:44 - Go MAEDA

- Affected version changed from 5.1.2 to 4.1.0

#6 - 2024-04-25 02:44 - Go MAEDA

- Related to Feature #29482: Query system for Projects page added

#7 - 2024-04-25 08:43 - Go MAEDA

- File 40610-v2.patch added

I found a simpler solution than the patch I posted in [#note-4](#). Applying the following fix will have the same effect as the patch in [#note-4](#), as it will skip processing inline images, which is not necessary when displaying the project list.

#8 - 2024-04-25 22:51 - James H

Thank you, will give it a try later

#9 - 2024-04-30 09:03 - Go MAEDA

- Target version changed from Candidate for next minor release to 5.0.9

Setting the target version to 5.0.9.

Files

40610.patch	2.2 KB	2024-04-25	Go MAEDA
40610-v2.patch	1009 Bytes	2024-04-25	Go MAEDA