

Regex Text on Receiver Email

2022/05/09 14:49 - Admin Redmine

ステータス:	Closed	開始日:	2010/07/10
優先度:	通常	期日:	
担当者:		進捗率:	0%
カテゴリ:	Email receiving_29	予定工数:	0.00時間
対象バージョン:	3.4.0_119	作業時間:	0.00時間
Redmineorg_URL:	https://www.redmine.org/issues/5864	status_id:	5
category_id:	29	tracker_id:	2
version_id:	119	plus1:	10
issue_org_id:	5864	affected_version:	
author_id:	17152	closed_on:	
assigned_to_id:	1	affected_version_id:	
comments:	37		

説明

Modify the "Truncate Email after these lines" section or add a new section that allows a regex to be added.. This would make the truncation of emails much easier and the ability for anyone to support their specific requirements for truncating after the email system replied to text ("On Wed June 24th 2010 Johnny wrote:").

This is related to #2852

I modified the controller for the mail handler to look for both the text truncation and hard coded the reply line that my email system uses.. It could easily be adapted to use a similar approach and regex check multiple expressions to handle different scenarios.

My Hard Coded Example:

```
regex = Regexp.new("(" + Regexp::MULTILINE.join('|') + ")s*[\r\n]. )/(On (.)wrote:[\r\n].*)", Regexp::MULTILINE)
```

A proposed generic pseudo-code controller example:

```
regex = Regexp.new("(" + Regexp::MULTILINE.join('|') + ")s*[\r\n].*)|(" + Regexp::MULTILINE.join('|') + ")", Regexp::MULTILINE)
```

journals

+1

+1

Sam, where did you put this? I'm looking in mail_handler_controller.rb and that doesn't look like the correct place to put this. I also want to be able to regex out a certain portion of all incoming emails. I use google apps/smtp for all received emails.

Sam Bo wrote:

My Hard Coded Example:

```
regex = Regexp.new("(" + Regexp::MULTILINE.join('|') + ")s*[\r\n]. )/(On (.)wrote:[\r\n].*)", Regexp::MULTILINE)
```

A proposed generic pseudo-code controller example:

```
regex = Regexp.new("(" + Regexp::MULTILINE.join('|') + ")s*[\r\n].*)|(" + Regexp::MULTILINE.join('|') + ")", Regexp::MULTILINE)
```

Akshat, take a here: source:trunk/app/models/mail_handler.rb#L321

I'm going to +1 (not the G+ kind mind you) this since it has been over a year. I'd love to see this so we don't have to manually update the model on each upgrade! If I knew anything about Ruby I'd submit a patch.

Ouch, the plus 1 didn't come through correctly. Sorry for the double posting here.

```
diff -r 146377aeb5a8 app/models/mail_handler.rb
--- a/app/models/mail_handler.rb      Sun May 13 19:09:35 2012 +0000
+++ b/app/models/mail_handler.rb      Fri May 25 23:00:22 2012 +0000
@@ -415,9 +415,9 @@
# Removes the email body of text after the truncation configurations.
def cleanup_body(body)
-  delimiters = Setting.mail_handler_body_delimiters.to_s.split(/[\r\n]+/).reject(&:blank?).map {|s| Regexp.escape(s)}
+  delimiters = Setting.mail_handler_body_delimiters.to_s.split(/[\r\n]+/).reject(&:blank?)
  unless delimiters.empty?
-    regex = Regexp.new("^[> ]*({ delimiters.join('|')})*s*[\r\n].*", Regexp::MULTILINE)
+    regex = Regexp.new("^(#{ delimiters.join('|')})*s*[\r\n].*", Regexp::MULTILINE)
    body = body.gsub(regex, "")
  end
  body.strip
```

----- +1 This would be extremely helpfull for filtering multilingual email replys.
Depending on the users email programm, the sender could be displayed as: Von: tickets@example.com
[mailto:tickets@example.com] or From: tickets@example.com [mailto:tickets@example.com] With a regular expression, this could
be filtered out with a single line in the redmine configuration. ----- The use of
multiline regex can make it quite tricky to write delimiters correctly. e.g. if you start your delimiter with @.*@ then it can match the
whole message, resulting in the whole message being deleted. I went for a slightly safer fix: use a normal regex, and if you find a
line matching any of the delimiters, delete that line and anything after it.

```
diff --git a/app/models/mail_handler.rb b/app/models/mail_handler.rb
index c84672b..82fd5fe 100644
--- a/app/models/mail_handler.rb
+++ b/app/models/mail_handler.rb
@@ -483,10 +483,16 @@
@@ -483,10 +483,16 @@ class MailHandler < ActionMailer::Base

# Removes the email body of text after the truncation configurations.
def cleanup_body(body)
-  delimiters = Setting.mail_handler_body_delimiters.to_s.split(/[\r\n]+/).reject(&:blank?).map {|s| Regexp.escape(s)}
+  delimiters = Setting.mail_handler_body_delimiters.to_s.split(/[\r\n]+/).reject(&:blank?)
  unless delimiters.empty?
-    regex = Regexp.new("^[> ]*({ delimiters.join('|')})*s*[\r\n].*", Regexp::MULTILINE)
-    body = body.gsub(regex, "")
+    # Combine all delimiters into one regex
+    regex = Regexp.new("^(#{ delimiters.join('|')})*")
+
+    # If the regex matches a line
+    regex.match(body) { |m|
+      # Delete the matched line and everything after it
+      body = body[0 ... m.begin(0)]
+    }
  end
  body.strip
end
```

----- Chris Birchall wrote: > The use of multiline regex can make it quite tricky to
write delimiters correctly. e.g. if you start your delimiter with @.*@ then it can match the whole message, resulting in the whole
message being deleted. > > I went for a slightly safer fix: use a normal regex, and if you find a line matching any of the delimiters,
delete that line and anything after it. > > [...] This worked perfectly. Thank you, Chris!

----- The patch by Chris
worked beautifully in Redmine 2.3.2. Thanks! +1 ----- I did not want to maintain
a patch for my installation of Redmine, so I've created a plugin (tested only on 2.4.3 thus far) changing the behavior of the
truncation field from text snippets per line to regular expressions per line.
http://www.redmine.org/plugins/redmine_mail_handler_clean_body_regexp

----- +1 Thank you for this life saving patch, the plugin works like a charm!
----- +1 ----- +1
----- +1 I'd so much like to see this integrated in redmine!
----- +1, nice feature. ----- +1,
please do ----- +10! -----
Thanks for this! I nearly have email truncation working as I'd like to - ie. it wasn't working at all with stock @3.2.0@ redmine for

me. However, with the above tweak of @app/models/mail_handler.rb@ it finally actually performs some truncation. Now, the last thing I'd like to do is get rid of email clients' first line on replies (as mentioned by Jens Schneider), ie.

On Mon, Feb 1, 2016 at 12:35, redmine@foo.com wrote:

```
> Feature #5864: Regex Text on Receiver Email
> blabla
> blabblabla
```

So I'm looking for how to write a @regex@ that would find & select the first line where it finds the mention of the redmine server's email address - so as to delete it, as well as any line following it.

For now I have come up with: @(redmine@foo.com)@

Which sort of works ok, when I test it with a sample text - here: <http://rubular.com/r/OowzlArxPf>

But it's not working when I declare it in redmine..

Is it because the regex is not good (enough)?

Or is it because I'm not writing/putting it in correctly in redmine's Admin interface? Should something be prefixed|appended to the regex for it to be taken into account?

!redmine_regex_incoming_emails.png!

For anyone interested, I finally have configured my redmine 3.2.0 to cleanly truncate incoming emails.

h3. Amend mail_handler.rb

The default @mail_handler.rb@ was not performing any truncation for me (not sure why; never got an answer/much help to try and troubleshoot it, cf. #21746).

So, reading the thread of comments on #5864, I finally found that the following works:

```
def cleanup_body(body)
  delimiters = Setting.mail_handler_body_delimiters.to_s.split(/[\r\n]+/).reject(&:blank?)
  unless delimiters.empty?
    # Combine all delimiters into one regex
    regex = Regexp.new("^> ]*#{ delimiters.join('|') }")
    # If the regex matches a line
    regex.match(body) { |m|
      # Delete the matched line and everything after it
      body = body[0 ... m.begin(0)]
    }
  end
  body.strip
end
```

h3. Administration / Settings / Incoming Emails In redmine's incoming emails settings, I then put the following values:

```
.+redmine@foo\.com.+
^-{2}.\n
.+image:\scid:.+
```

The first one, @.+redmine@foo.com.+@, enables to wipe out email clients' top line on replies, such as:

On Mon, Feb 1, 2016 at 12:35, redmine@foo.com wrote:

```
> Feature #5864: Regex Text on Receiver Email
> blabla
> blabblabla
```

So that's very cool.

The other two settings are:

- @-^{2}.\n@ is to remove Gmail's appended signatures, which apparently are always preceded by a line with @--@ (followed by an invisible single character; which when you look at Gmail.com's online @raw email dump@ feature, they show as being @=20@,

but it isn't, it's a single character of whateverz they append to that double tack, hence the "." used in the regex).

- Whereas @.+image:\scid:.\+@ is to remove an image logo our company has for our staff's email signature.

Also, for that signature image, I also declared the full file name, in the *@Exclude attachments by name@* setting.

I might have to add file exclusions if/when we allow non-staff, ie. clients, to create/comment issues via email, but for now this setup works flawlessly!

h3. Caveats

I've noticed that the regex truncation and/or attachment exclusion rules I specify in redmine's Administration are not taken into account upon *@Save@*. I have to restart redmine application, and then they're properly taken into account.

I'm not sure if that's normal; but mentionning it in case someone else reads this, as it'll save you a lot of time and pain.

If someone knows that this is a normal behaviour, then we should probably add a notice in Administration saying "Please restart redmine application for these settings to be taken into account".

If it's not normal, please let me know, and I can provide info on server setup (nginx+passenger+rben4rubies basically).

h3. Final note

I tried using the "redmine_mail_handler_clean_body_regex plugin":

http://www.redmine.org/plugins/redmine_mail_handler_clean_body_regex - it didn't work for me. Not sure if because I'm running redmine 3.2.0.

We made a patch with tests that implements this feature. Without regex delimiters, we weren't able to truncate correctly the emails received. I think that this feature is really needed in core.

Marius BALTEANU wrote:

We made a patch with tests that implements this feature. Without regex delimiters, we weren't able to truncate correctly the emails received. I think that this feature is really needed in core.

Does this patch break if existing setting has regex special characters?

I think it is better to switch regex on or off.

Toshi MARUYAMA wrote:

Marius BALTEANU wrote:

We made a patch with tests that implements this feature. Without regex delimiters, we weren't able to truncate correctly the emails received. I think that this feature is really needed in core.

Does this patch break if existing setting has regex special characters?

From my tests, no, it doesn't break, but if you have a specific scenario in your mind, please tell me and I'll test it.

I think it is better to switch regex on or off.

Yes, agree with you. I've updated the patch to add a new setting which enable/disable this feature.

!regex_delimiter_setting.png!

Excellent patch, thanks for contributing it!

Marius BALTEANU wrote:

Toshi MARUYAMA wrote:

Marius BALTEANU wrote:

We made a patch with tests that implements this feature. Without regex delimiters, we weren't able to truncate correctly the emails received. I think that this feature is really needed in core.

Does this patch break if existing setting has regexp special characters?

From my tests, no, it doesn't break, but if you have a specific scenario in your mind, please tell me and I'll test it.

For example, "*cut below lines*".

Toshi MARUYAMA wrote:

For example, "*cut below lines*".

You're right, without the new setting to enable/disable this feature, the existing delimiters with special regex characters will behave differently.

I think that this patch would raise an error when receiving an email if "Enable regexp delimiters" is checked and the entered delimiter is an invalid regexp.

Jean-Philippe Lang wrote:

I think that this patch would raise an error when receiving an email if "Enable regexp delimiters" is checked and the entered delimiter is an invalid regexp.

Thanks for your feedback. I'll modify the patch to validate each regex on save when the "Enable regexp delimiters" is checked.

I've updated the patch to validate each regex delimiter on save. The user won't be able to save new settings with invalid regex delimiters and will receive an error message for each invalid entry.

Please let me know if more changes are required to have this committed.

Thanks for implementing this feature. Attached is a small patch that makes the text "Enable regular expressions" clickable (as a label for the checkbox).

I've fixed it using an existing class, thanks for pointing this out.

related_issues

relates,Closed,11684,Truncate incoming emails
duplicates,Closed,10069,delimiter improvments

履歴

#1 - 2022/05/10 17:22 - Admin Redmine

- カテゴリを Email receiving_29 にセット
- 対象バージョンを 3.4.0_119 にセット