

Redmine - Defect #20192

IssueLink in email notification contains SUB-URI part twice

2015-06-29 07:17 - Miklós Kriván

Status:	Needs feedback	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Email notifications	Estimated time:	0.00 hour
Target version:		Affected version:	3.0.3
Resolution:			

Description

I am using Redmine installed in SUB-URI environment for a long time now I plan to upgrade from 2.5.3 into 3.0.3. So I have started testing it before I upgrade my production environment. Everything looks good only one problem I have recognized.

When I receive an email notification from Redmine 3.0.3 installed in sub-uri environment, the URL contains the SUB-URI part twice like this <http://server.domain/redmine/redmine/issues/1234#change-12345>

All of the other URL-s are working properly in my Redmine environment.

My general site settings contains only server.domain without the sub-uri part.

If I use Redmine 2.5 or 2.6 versions with exactly the same setup everything work as expected.

I have installed everything with SVN checkout and the suggested building procedure.

All of the functionality in redmine works perfect only the URLs in email notifications are wrong (SUB-URI part doubled).

I have checked the app/models/mailer.rb file and I have found an important change compared to earlier versions:

In 2.5 and 2.6 it was:

```
def self.default_url_options
  { :host => Setting.host_name, :protocol => Setting.protocol }
end
```

In 3.0.3 it is:

```
def self.default_url_options
  options = {:protocol => Setting.protocol}
  if Setting.host_name.to_s =~ /\A(https?\:\:\//)?(.+?) (\:(\d+))?(\/\.)?\z/i
    host, port, prefix = $2, $4, $5
    options.merge!({
      :host => host, :port => port, :script_name => prefix
    })
  else
    options[:host] = Setting.host_name
  end
  options
end
```

I have changed the :script_name attribute in the 31st line like this:

```
:script_name => ''
```

So the function now looks like:

```
def self.default_url_options
  options = {:protocol => Setting.protocol}
  if Setting.host_name.to_s =~ /\A(https?\:\:\//)?(.+?) (\:(\d+))?(\/\.)?\z/i
    host, port, prefix = $2, $4, $5
    options.merge!({
      :host => host, :port => port, :script_name => ''
    })
  end
end
```

```
else
  options[:host] = Setting.host_name
end
options
end
```

And the problem is solved but I would like to be sure is this solution does not generates other unexpected behaviour.

My environment is:

Environment:

```
Redmine version      3.0.3.stable.14377
Ruby version         2.0.0-p0 (2013-02-24) [x86_64-linux]
Rails version        4.2.3
Environment          production
Database adapter     Mysql2
```

SCM:

```
Subversion           1.6.9
Git                  1.7.11.3
```

Filesystem

Redmine plugins:

```
clipboard_image_paste 1.10
quick_edit             0.0.8
redmine_agile          1.3.9
redmine_ckeditor       1.1.0
redmine_dashboard     2.5.0
redmine_helpdesk      0.0.12
redmine_my_page        0.1.5
redmine_per_project_formatting 0.0.4
redmine_theme_changer 0.2.0
redmine_wiki_unc       0.0.3
redmine_work_time      0.3.0
redmine_xlsx_format_issue_exporter 0.1.2
```

Thx. for your help

Related issues:

Related to Redmine - Defect #19323: Incorrect links generated in emails if ho...	Closed
Has duplicate Redmine - Defect #23182: url in email + sub uri = sub uri dupli...	Closed

History

#1 - 2015-06-29 07:26 - Miklós Kriván

This is my apache configuration:

```
FastCgiServer /data/redmine/public/dispatch.fcgi \
  -idle-timeout 120 -processes 2 \
  -initial-env RAILS_ENV=production \
  -initial-env PATH=/data/ruby/bin:/usr/bin \
  -initial-env LD_LIBRARY_PATH=/data/tools/lib \
  -initial-env RAILS_RELATIVE_URL_ROOT=/redmine
```

```
Alias /redmine /data/redmine/public
<Directory /data/redmine/public>
  Options Indexes ExecCGI FollowSymLinks
```

```
RewriteEngine On
RewriteBase /redmine
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^(.*)$ dispatch.fcgi
```

```
AllowOverride all
Order allow,deny
Allow from all
</Directory>
```

in redmine/config/environment.rb

```
RedmineApp::Application.routes.default_scope = {
  :path => ENV['RAILS_RELATIVE_URL_ROOT'],
```

```
:shallow_path => ENV['RAILS_RELATIVE_URL_ROOT']
}
```

#2 - 2015-06-29 07:33 - Toshi MARUYAMA

- Target version set to 3.0.4

It seems [#19323](#) regression.

#3 - 2015-06-29 07:34 - Toshi MARUYAMA

- Related to Defect #19323: Incorrect links generated in emails if host setup uses other port (":" symbol) added

#4 - 2015-06-29 19:29 - Jean-Philippe Lang

I've made a test with your change to environment.rb, and don't get the double sub-uri if I set the "Host name and path" (in application settings / General tab) to the host name only, without the sub-uri path (eg. redmine.host). Can you double check that?

#5 - 2015-07-07 21:40 - Jean-Philippe Lang

- Target version changed from 3.0.4 to Candidate for next minor release

#6 - 2015-07-25 05:38 - Toshi MARUYAMA

- Status changed from New to Needs feedback

- Target version deleted (Candidate for next minor release)

#7 - 2016-02-24 18:20 - Robert Schneider

I can confirm the problem. The workaround does help me!

Though I think I don't have the same situation. Hard to describe what my situation is, actually... ;-) Has something to do with passenger. At least I can say all worked fine except the emails. With this hack it also works for me with Redmine 3.1.0 and 3.2.0.

#8 - 2016-02-25 08:53 - Robert Schneider

I'm wondering why prefix (or :script_name) gets populated. In my case Setting.host_name is set to redmine.colop.com, so there is no /something appended to the host. Links should be redmine.colop.com/something but actually get rendered to redmine.colop.com/something/something.

In my case config/additional_environment.rb and config/environments/production.rb are not modified. relative_url_root is not changed that way. And my httpd-conf contains:

```
<VirtualHost *:80>
  ServerName redmine.colop.com

  PassengerRuby /home/redmine/.rvm/wrappers/ruby-2.2.1/ruby
  PassengerSpawnMethod smart
  PassengerPoolIdleTime 300
  PassengerMaxRequests 500
  PassengerStatThrottleRate 5
  PassengerMinInstances 2

  DocumentRoot /var/www/html/redmine/public

  Alias /something /var/www/html/redmine/public
  <Location "/something">
    PassengerBaseURI /something
    PassengerAppRoot /var/www/html/redmine
  </Location>

  <Directory /var/www/html/redmine/public>
    Allow from all
    Options -MultiViews
    # Uncomment this if you're on Apache >= 2.4:
    #Require all granted
  </Directory>
</VirtualHost>
```

Does this configuratin inject the host into Redmine configuration? Why can it happen that :script_name is not empty?

#9 - 2016-06-28 04:33 - Toshi MARUYAMA

- Has duplicate Defect #23182: url in email + sub uri = sub uri duplication added

#10 - 2016-06-28 08:23 - Evgeniy Dushistov

Jean-Philippe Lang wrote:

I've made a test with your change to environment.rb, and don't get the double sub-uri if I set the "Host name and path" (in application settings / General tab) to the host name only, without the sub-uri path (eg. redmine.host). Can you double check that?

At first time I setted
hostname to "host/redmine" in redmine's preferences
(as redmine suggest, it show under input caption: "Example: host/redmine").

Then, when I saw URL like this "host/redmine/redmine" in mail,
so I changed hostname to just "host", but emails still contains double sub-uri in URL.

But hostname change works, if for example I set hostname to "host/" I got in email
"host//redmine/redmine", as you see it contains "//".

So may be to reproduce bug, you need at first change hostname to "host/redmine",
and then to "host".

#11 - 2016-06-28 08:53 - Evgeniy Dushistov

Indeed

```
:script_name => ''
```

fixed issue for me.

I add some logging into app/models/mailer.rb:

```
def self.default_url_options
  Rails.logger.error "Hi!!!!!! It is me!!!!!!!!!!!!!!!!!!!!!!"
  options = {:protocol => Setting.protocol}
  Rails.logger.error "!!!!!!Setting.host_name.to_s '#{Setting.host_name.to_s}'"
  if Setting.host_name.to_s =~ /\A(?:https?\:\/\/)?(.*?) (\:(\d+))?(\/.*)?\/z/i
    host, port, prefix = $2, $4, $5

    Rails.logger.error "!!!!!! '#{host}', '#{port}', '#{prefix}'"
    options.merge!({
      :host => host, :port => port, :script_name => ''
    })
  else
    Rails.logger.error "!!!!!!ELSE!!!! Setting.host_name '#{Setting.host_name}'"
    options[:host] = Setting.host_name
  end
  options
end
```

It prints:

```
Hi!!!!!! It is me!!!!!!!!!!!!!!!!!!!!!!
!!!!!!Setting.host_name.to_s 'hostname.net'
!!!!!! 'hostname.net', '', ''
```

So prefix is empty, but not as empty string.

Because of

```
:script_name => prefix
```

and

```
:script_name => ''
```

give different results.