

Redmine - Defect #35715

File upload fails when run with uWSGI

2021-08-06 10:50 - Cyprien D

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Go MAEDA	% Done:	0%
Category:	Attachments	Estimated time:	0.00 hour
Target version:	4.1.5	Affected version:	4.2.2
Resolution:	Fixed		
Description			
Hello,			
Attachments are broken since I have upgrade my Redmine from 4.2.1 to 4.2.2. POST fail with a 500 error and this error is logged:			
<pre>Started POST "/uploads.js?attachment_id=2&filename=placeholder.jpeg&content_type=image%2Fjpeg" for X.X.X.X at 2021-08-05 17:48:28 +0200 Processing by AttachmentsController#upload as JS Parameters: {"attachment_id"=>"2", "filename"=>"placeholder.jpeg", "content_type"=>"image/jpeg"} Current user: cyprien (id=121) Completed 500 Internal Server Error in 5ms (ActiveRecord: 0.9ms) NoMethodError (undefined method `size' for #<Uwsgi_IO:0x00005566ba40c8d0>): app/models/attachment.rb:123:in `file=' app/controllers/attachments_controller.rb:108:in `upload' lib/redmine/sudo_mode.rb:61:in `sudo_mode'</pre>			
It seems to come from this change introduced to 4.2.2			
<pre>diff redmine-4.2.1/app/controllers/attachments_controller.rb redmine-4.2.2/app/controllers/attachm ents_controller.rb 108c108 < @attachment = Attachment.new(:file => request.raw_post) --- > @attachment = Attachment.new(:file => request.body)</pre>			
If I switch back from request.body to request.raw_post the error is fixed			
Related issues:			
Related to Redmine - Defect #33752: Uploading a big file fails with NoMemoryE...			Closed

Associated revisions

Revision 21173 - 2021-08-14 03:38 - Go MAEDA

File upload fails when run with uWSGI (#35715).

Contributed by Pavel Rosický and Holger Just.

Revision 21174 - 2021-08-15 06:17 - Go MAEDA

Merged r21173 from trunk to 4.2-stable (#35715).

Revision 21175 - 2021-08-15 06:19 - Go MAEDA

Merged r21173 from trunk to 4.1-stable (#35715).

History

#1 - 2021-08-06 17:05 - Pavel Rosický

I'm not sure if uwsgi is officially supported. The best way would be if they implement the method in their adapter.

https://github.com/unbit/uwsgi/blob/master/plugins/rack/rack_plugin.c

we could use a fallback, but it'll load the whole file into a memory again on adapters that don't support the feature. A compromise?

```
diff --git a/app/controllers/attachments_controller.rb b/app/controllers/attachments_controller.rb
index 8e28194a3..f62c35976 100644
--- a/app/controllers/attachments_controller.rb
+++ b/app/controllers/attachments_controller.rb
@@ -105,7 +105,7 @@ class AttachmentsController < ApplicationController
   return
   end

-   @attachment = Attachment.new(:file => request.body)
+   @attachment = Attachment.new(:file => request.body.respond_to?(:size) ? request.body : request.raw_post)
   @attachment.author = User.current
   @attachment.filename = params[:filename].presence || Redmine::Utils.random_hex(16)
   @attachment.content_type = params[:content_type].presence
```

Go MAEDA what do you think?

#2 - 2021-08-11 21:30 - Holger Just

- *Related to Defect #33752: Uploading a big file fails with NoMemoryError added*

#3 - 2021-08-11 21:48 - Holger Just

uwsgi is certainly an unusual application server... However, as restoring basic support for it should be rather simple, and it did work before, I'm in favor of providing a fallback as you proposed.

I'd just recommend to extract the check to a separate method with some comments describing why we are doing this. This could look like:

```
class AttachmentController
  #...

  private

  # Get an IO-like object for the request body which is usable to create a new
  # attachment. We try to avoid having to read the whole body into memory.
  def raw_request_body
    if request.body.respond_to?(:size)
      request.body
    else
      request.raw_body
    end
  end
end
```

#4 - 2021-08-12 06:43 - Go MAEDA

- *File 35715.patch added*

- *Target version set to 4.2.3*

Pavel and Holger, thank you for looking into the issue and writing the patch. I will commit [35715.patch](#).

Setting the target version to 4.2.3.

#5 - 2021-08-12 09:54 - Pavel Rosický

thanks, I agree, but the target version should be 4.1.5 because the original change went to 4.1.4

and there's also a typo in the new patch, it should be request.raw_post instead of request.raw_body

#6 - 2021-08-12 16:21 - Marius BĂLTEANU

- *Target version changed from 4.2.3 to 4.1.5*

#7 - 2021-08-13 02:55 - Go MAEDA

- *File 35715-v2.patch added*

Pavel Rosický wrote:

and there's also a typo in the new patch, it should be `request.raw_post` instead of `request.raw_body`

Fixed the patch. And I added a test that simulates a request body without the size method. Thank you for pointing it out.

#8 - 2021-08-14 03:38 - Go MAEDA

- *Subject changed from Attachments fails - 500 Internal Server Error on uploads.js POST to File upload fails when run with uWSGI*
- *Status changed from New to Resolved*
- *Assignee set to Go MAEDA*
- *Resolution set to Fixed*

Committed the fix.

Thank you all for reporting and fixing the issue.

#9 - 2021-08-15 06:19 - Go MAEDA

- *Status changed from Resolved to Closed*

Files

35715.patch	1.09 KB	2021-08-12	Go MAEDA
35715-v2.patch	2.44 KB	2021-08-13	Go MAEDA