

## Redmine - Feature #7725

### REST API for files

2011-02-25 21:56 - Alex Last

<b>Status:</b>	Closed	<b>Start date:</b>	2011-02-25
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>	REST API	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>			
<b>Resolution:</b>	Duplicate		
<b>Description</b> would be nice to have the API to manipulate files remotely. e.g. to upload new distributives using some automated scripts.			
<b>Related issues:</b> Related to Redmine - Feature #18245: Documents REST API <b>New</b> Related to Redmine - Feature #13800: File REST API <b>New</b> Related to Redmine - Patch #19116: Files REST API <b>Closed</b> Has duplicate Redmine - Feature #20404: REST API support for project files (a... <b>Closed</b>			

### History

#### #1 - 2011-05-16 13:48 - j dlx

Alexey Skor wrote:

would be nice to have the API to manipulate files remotely.  
e.g. to upload new distributives using some automated scripts.

Well for starters even a simple *files* list would do for me..  
right now the only way (i know of) is to filter *activity*..

May i ask the maintainer/some official to get some word out on the files-API topic?

Reading through all/most threads regarding this topics, i still have no real clue if:

1. such a feature is planned (or at least considered) at all
2. if so, if there's a roadmap or something?

thx,  
Jan

#### #2 - 2012-03-13 12:08 - Gert van Dijk

Lately, some progress is made on the REST API features like the file attachments - see [#8171](#). However, as far as I can see from the current documentation, it's still not possible to manage a project's 'Files' and 'Documents' using the REST API.

I would also like to see this feature for automated uploading of files; e.g. test builds after automated building in a post-receive/post-commit hook.

#### #3 - 2012-03-18 21:18 - Gert van Dijk

In the meantime I created a Python script to overcome this. See [redmine-file-uploader on GitHub](#).

I can now upload my builds from the build server to Redmine in the 'Files' section of the project for internal testing/deployment.  
For anyone also in need of this; hope this helps.

#### #4 - 2013-05-12 17:04 - Bengt Lüers

The linked script works fine for uploading, but I would like to download files to mirror them in a Git repository. Is there any chance that this will make it into the API, soon or do I have to write something that does the downloading in the same fashion?

#### #5 - 2013-06-27 11:24 - Dmitry Popov

Thanks, Gert! Your script works like a charm!

But still, API support would be appreciated.

## #6 - 2013-06-27 15:49 - Dmitry Popov

I've rewritten the script in Ruby:

```
require 'mechanize'

# Replaces \ with / and removes "
ARGV.map!{|a|a.gsub('\','/').gsub(/^(.+) "$/, '\1')}

filename = ARGV[0] || abort('Filename must be specified')
puts "File:      #{filename}"
url       = ARGV[1] || abort('Redmine URL must be specified')
puts "Redmine URL:  #{url}"
username  = ARGV[2] || abort('Redmine username must be specified')
puts "Username:    #{username}"
password  = ARGV[3] || abort('Redmine password must be specified')
puts "Password:    #{'*' * password.length}"
project   = ARGV[4] || abort('Redmine project must be specified')
puts "Project:     #{project}"

login_page_path = '/login'
files_page_path = '/projects/' + project + '/files'

agent = Mechanize.new

# No certificate verification (I had to use this hack because our server is bound to custom port)
# agent.agent.http.verify_mode = OpenSSL::SSL::VERIFY_NONE

agent.get(URI.join(url, login_page_path)) do |login_page|
  login_page.form_with(:action => login_page_path) do |login_form|
    login_form.username = username
    login_form.password = password
  end.submit
end

agent.get(URI.join(url, files_page_path + '/new')) do |upload_page|
  upload_page.form_with(:action => files_page_path) do |upload_form|
    upload_form.file_uploads.first.file_name = filename
  end.submit
end
```

Here's the script to delete uploaded files:

```
agent.get(URI.join(url, files_page_path)) do |files_page|
  file_link = files_page.link_with(:text => File.basename(filename))
  if file_link
    file_id = file_link.href.match(/download\/(\d+)/)[1].to_i
    agent.post(URI.join(url, "/attachments/#{file_id}"), {'_method' => 'delete', 'authenticity_token' =>
files_page.at('meta[@name="csrf-token"]')[0].content})
  end
end
```

## #7 - 2015-02-11 17:41 - Lucile Quirion

I also need a REST API for uploading files and noted that there is several open tickets concerning that feature ([#18245](#), [#13800](#))

So I've started working on it.

I'll try to include the patch submitted in [#13800](#)

## #8 - 2015-03-12 07:35 - Toshi MARUYAMA

- Related to Feature #18245: Documents REST API added

## #9 - 2015-03-12 07:36 - Toshi MARUYAMA

- Related to Feature #13800: File REST API added

## #10 - 2015-03-18 08:57 - Toshi MARUYAMA

- Related to Patch #19116: Files REST API added

## #11 - 2015-07-25 06:26 - Mischa The Evil

- Has duplicate Feature #20404: REST API support for project files (attachments) added

**#12 - 2017-01-09 13:45 - Go MAEDA**

- *Status changed from New to Closed*
- *Resolution set to Duplicate*

Implemented by [#19116](#), will be available in upcoming 3.4.0.