

Redmine - Defect #35885

the change of routing raw files from repositories not included in the upgrade proces/manual

2021-09-17 16:14 - Wim Bertels

| | | | |
|---|--------|--------------------------|-----------|
| Status: | New | Start date: | |
| Priority: | Normal | Due date: | |
| Assignee: | | % Done: | 0% |
| Category: | | Estimated time: | 0.00 hour |
| Target version: | | Affected version: | 4.0.7 |
| Resolution: | | | |
| Description | | | |
| Hello, | | | |
| after upgrading, as described on https://wiki.debian.org/Redmine from version 3.3.1 to 4.0.7 | | | |
| i noticed that redmine has a another http routing to the raw download files of a named repository, i didn't find any documentation on this, maybe i missed it. | | | |
| for example, | | | |
| now this links works, (it is the download link presented in the redmine repository for this file): | | | |
| https://www.redmine.org/projects/redmine/repository/raw/sandbox/custom_fields/Gemfile | | | |
| (this repository has no explicit name identifier set on the settings page i'm guessing) | | | |
| <ul style="list-style-type: none">• in version 3.3.1 this same link would appear, whether or not the repository has a name identifier• in version 4.0.7 this same link would appear if the repository has no name identifier, otherwise if the name identifier is included as such for identifier XYZ: https://www.redmine.org/projects/redmine/repository/XYZ/raw/sandbox/custom_fields/Gemfile | | | |
| So after upgrading, none of the links referring to repository with a name identifiers still work, as they all point to https://.../projects/.../repository/raw/ instead of https://.../projects/.../repository/repo_name_identifier/raw/ . | | | |
| hth, suggestions welcome | | | |
| Related issues: | | | |
| Related to Redmine - Patch #26522: Repository routing bug when file path star... | | Closed | |

History

#1 - 2021-09-21 15:22 - Holger Just

- Related to Patch #26522: Repository routing bug when file path starts with (browse|entry|raw|changes|annotate|diff)/ added

#2 - 2021-09-21 15:27 - Holger Just

Unfortunately, the previous routing implementation could lead to the wrong route being selected if there was e.g a directory named raw in the default repository.

As such, the routing behavior was changed in [4.0.0](#) with [#26522](#) to remove the routes without an explicit repository id and to make sure we always select the correct action and repo path.

#3 - 2021-09-22 01:44 - Mischa The Evil

Wim Bertels wrote:

[...]

So after upgrading, none of the links referring to repository with a name identifiers still work, as they all point to <https://.../projects/.../repository/raw/> instead of https://.../projects/.../repository/repo_name_identifier/raw/.

Holger Just wrote:

[...]

As such, the routing behavior was changed in [4.0.0](#) with [#26522](#) to remove the routes without an explicit repository id and to make sure we always select the correct action and repo path.

Just for the clarity on this matter, I presume that Redmine links ('source:', 'export:') were not affected by this change (i.e. they link to the new routes) as opposed to direct links which are affected as reported in this issue.

With that being said, it shouldn't be very difficult to restore the prior behavior by adding the removed routes to a custom plugin in case you are sure that you won't be affected by the issue reported in [#26522](#).

#4 - 2021-09-22 10:14 - Wim Bertels

Mischa The Evil wrote:

[...]

Just for the clarity on this matter, I presume that Redmine links ('source:', 'export:') were not affected by this change (i.e. they link to the new routes) as opposed to direct links which are affected as reported in this issue.

With that being said, it shouldn't be very difficult to restore the prior behavior by adding the removed routes to a custom plugin in case you are sure that you won't be affected by the issue reported in [#26522](#).

that is helpful, i guess there is no way to hide 'source:' or 'export:' from the endviewer (as one can do with [link to redmine](#))

--

i have looked at the db schema: it seems there a table wiki_contents and wiki_content_versions that hold the actual content of the wiki pages.

do you think it would be a problem using a direct UPDATE statement to change the affected links in the table wiki_contents

small test done:

1. made a fresh wikipage with some content,
2. UPDATE wiki_content_version table of that fresh wikipage

result:

1. only 1 version in wiki_content_versions as expected
2. different content in wiki_contents
3. this different content gets shown on the wiki-website, but can't be found in the history

continu:

1. edit this wikipage from redmine, a second version appears, the previously UPDATED content (together with this edit) is archived in the history as 1 version together

this behaviour is fine for my usecase, or do you expect problems with this approach?

#5 - 2021-09-24 17:32 - Wim Bertels

If someone else experiences the same problem:

[fix_old_wiki_links_without_repo_identifier.sql](#) is a template for a script to change this in the database, use at own risk, no warranty

HTH