The right click context menu on issues is being submitted twice for each right click. You can see it right here on Redmine.org:

1. Go to issues list
2. Select some issues
3. Right click
4. Notice multiple POST requests sent in Firebug

I’m not sure but it looks like at the end of the Click function in context_menu.js, the Right Click is being called even though it’s already bound above. Can someone else confirm this issue for me?

- Browser: Firefox 3
- OS: Linux

Related issues:
Duplicated by Redmine - Defect # 5491: context_menu javascript POST'ed twice ...

Associated revisions
Revision 6195 - 2011-07-08 11:58 - Toshi MARUYAMA

Fixed: context menu is being submitted twice per right click (#2109).

Contributed by Emilio Carlos da Palma.

History
#1 - 2008-11-07 17:25 - Jean-Philippe Lang

I’m not able to reproduce this issue. Only one POST request is sent when right-clicking:

- Browser: Firefox 3
- OS: Win XP

LiveHTTPHeaders gives me:

GET /projects/redmine/issues

POST /projects/redmine/issues/context_menu ids%5B%5D=2143&ids%5B%5D=2139&ids%5B%5D=2134&ids%5B%5D=2133

Can you check with something else than Firebug please?
Jean-Philippe Lang wrote:

Can you check with something else than Firebug please?

- OS: Debian Linux
- Browser: Firefox 3 (Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.0.1) Gecko/2008071618 Iceweasel/3.0.1 (Debian-3.0.1-1))
- Redmine version: Trunk, r1990 (git: 63c72fe87f901a6a9f6f1981a4e93a366be05f)

Sure here is the development.log, production.log, and HTTPFox results.

# Here is the Rails development.log
# ...Lots of checking for helper lines here...

Processing IssuesController#context_menu (for 127.0.0.1 at 2008-11-07 09:47:07) [POST]

Session ID: 80269b90cd68b8e38a8a15a89af61e
Parameters: {"project_id":"test", "ids":["2", "1"], "action":"context_menu", "controller":"issues"}

Setting Columns (0.000590) SHOW FIELDS FROM `settings`

SQL (0.000072) SELECT max(`settings`.updated_on) AS max_updated_on FROM `settings`

AnonymousUser Columns (0.000977) SHOW FIELDS FROM `users`
AnonymousUser Load (0.000019) SELECT * FROM `users` WHERE ( {`users`.type = 'AnonymousUser' } ) LIMIT 1

Setting Load (0.000083) SELECT * FROM `settings` WHERE {`settings`.name = 'login_required' } LIMIT 1

Issue Columns (0.001069) SHOW FIELDS FROM `issues`
Issue Load (0.000381) SELECT * FROM `issues` WHERE (`issues`.id IN ("2","1"))

Project Load (0.000112) SELECT * FROM `projects` WHERE (`projects`.id IN ("1"))

Project Load (0.000277) SELECT * FROM `projects`

EnabledModule Load (0.000118) SELECT * FROM `enabled_modules` WHERE ( `enabled_modules`.project_id = 1)

Role Load (0.000133) SELECT * FROM `roles` WHERE ( `roles`.builtin = 2 ) LIMIT 1

Role Load (0.000622) SHOW FIELDS FROM `roles`

Member Columns (0.000711) SHOW FIELDS FROM `members`

User Columns (0.0001077) SHOW FIELDS FROM `users`

Member Load Including Associations (0.000491) SELECT `members`.id AS t0_r0, `members`.user_id AS t0_r1, `members`.project_id AS t0_r2, `members`.role_id AS t0_r3, `members`.created_on AS t0_r4, `members`.mail_notification AS t0_r5, `users`.id AS t1_r0, `users`.login AS t1_r1, `users`.hashed_password AS t1_r2, `users`.firstname AS t1_r3, `users`.lastname AS t1_r4, `users`.mail AS t1_r5, `users`.mail_notification AS t1_r6, `users`.admin AS t1_r7, `users`.status AS t1_r8, `users`.last_login_on AS t1_r9, `users`.language AS t1_r10, `users`.auth_source_id AS t1_r11, `users`.created_on AS t1_r12, `users`.updated_on AS t1_r13, `users`.type AS t1_r14 FROM `members` LEFT OUTER JOIN `users` ON `users`.id = `members`.user_id WHERE ( `members`.project_id = 1 AND ( `users`.status=1) )

Role Load (0.000137) SELECT * FROM `roles` WHERE ( `roles`.id = 3)

Enumeration Load (0.000123) SELECT * FROM `enumerations` WHERE ( `enumerations`.opt = 'IPRI' ) ORDER BY position

IssueStatus Load (0.000106) SELECT * FROM `issue_statuses` ORDER BY position

Rendering issues/context_menu

Enumeration Columns (0.000678) SHOW FIELDS FROM `enumerations`

Version Columns (0.000699) SHOW FIELDS FROM `versions`

Version Load (0.000069) SELECT count(*) AS count_all FROM `versions` WHERE (`versions`.project_id = 1)

IssueCategory Columns (0.000597) SHOW FIELDS FROM `issue_categories`

SQL (0.000119) SELECT count(*) AS count_all FROM `issue_categories` WHERE (`issue_categories`.project_id = 1)
Lots of checking for helper lines here...

Processing IssuesController#context_menu (for 127.0.0.1 at 2008-11-07 09:47:08) [POST]
  Session ID: 80269b9b0cd68bec83a8ae15a89af61e
  Parameters: {"project_id"=>"test", "ids=>"[2", "1]", "action"=>"context_menu", "controller"=>"issues"}
  Setting Columns (0.000737) SHOW FIELDS FROM `settings`
  SQL (0.000155) SELECT max(`settings`.updated_on) AS max_updated_on FROM `settings`
  AnonymousUser Columns (0.001102) SHOW FIELDS FROM `users`

# Here is the same application in production mode (bit easier to see the two requests):
#
# => Rails 2.1.0 application starting on http://0.0.0.0:3000
# => Debugger enabled
# => Call with -d to detach
# => Ctrl-C to shutdown server
** Starting Mongrel listening at 0.0.0.0:3000
** Starting Rails with production environment...
** Rails loaded.
** Signals ready. TERM => stop. USR2 => restart. INT => stop (no restart).
** Rails signals registered. HUP => reload (without restart). It might not work well.
** Mongrel 1.1.5 available at 0.0.0.0:3000
** Use CTRL-C to stop.

Processing IssuesController#index (for 127.0.0.1 at 2008-11-07 09:50:52) [GET]
  Session ID: 80269b9b0cd68bec83a8ae15a89af61e
  Parameters: {"project_id"=>"test", "action"="index", "controller"="issues"}
  Rendering template within layouts/base
  Rendering issues/index.rhtml
Completed in 0.59918 (1 reqs/sec) | Rendering: 0.18041 (30%) | DB: 0.01851 (3%) | 200 OK [http://localhost/projects/test/issues]

Processing IssuesController#context_menu (for 127.0.0.1 at 2008-11-07 09:50:58) [POST]
  Session ID: 80269b9b0cd68bec83a8ae15a89af61e
  Parameters: {"project_id"=>"test", "ids=>"[2", "1]", "action"="context_menu", "controller"="issues"}
  Rendering issues/context_menu
Completed in 0.06734 (14 reqs/sec) | Rendering: 0.03890 (57%) | DB: 0.00904 (35%) | 200 OK [http://localhost/projects/test/issues/context_menu]

Processing IssuesController#context_menu (for 127.0.0.1 at 2008-11-07 09:50:58) [POST]
  Session ID: 80269b9b0cd68bec83a8ae15a89af61e
  Parameters: {"project_id"=>"test", "ids=>"[2", "1]", "action"="context_menu", "controller"="issues"}
  Rendering issues/context_menu
Completed in 0.02551 (39 reqs/sec) | Rendering: 0.00651 (25%) | DB: 0.00904 (35%) | 200 OK [http://localhost/projects/test/issues/context_menu]

# Using HTTPFox I get this:
# Load issues page
00:00:07.770  0.154   455   (26475)   GET   304   text/html http://localhost:3000/projects/test/issues

2020-04-09
I can confirm the issue. We have the same problem when using Firefox (3.0.11) from Linux, while right click from Firefox (3.0.11) from Windows XP is sending only one request.

I commented out "this.RightClick(e);" which is called near the end of the Click()-Method, and it seems to work fine now (so far). I just don’t understand, what that call was meant for? Maybe it’s from an earlier version where you handled left and right clicks in one method?

Regards
Chris

This is a patch from https://www.chiliproject.org/issues/424

Firefox 3.5 calls twice.

$ rpm -q firefox
firefox-3.5.15-1.fc12.i686

Processing ContextMenusController#issues (for 192.168.11.7 at 2011-07-08 18:38:59) [GET]
.
.
Processing ContextMenusController#issues (for 192.168.11.7 at 2011-07-08 18:39:00) [GET]
Google chrome calls once.

```
$ rpm -q google-chrome-stable
google-chrome-stable-11.0.696.71-86024.i386
```

#6 - 2011-07-08 12:07 - Toshi MARUYAMA

- Status changed from New to Closed
- Target version set to 1.3.0
- Resolution set to Fixed

I confirmed note 4 patch works fine on Windows.
  - Firefox 3.5.19
  - IE 8.0.6001.19088
  - Google chrome 12.0.742.112

Files

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>context-menu.diff</td>
<td>519 Bytes</td>
<td>2011-05-28</td>
<td>Toshi MARUYAMA</td>
</tr>
</tbody>
</table>