

Redmine - Patch #5717

Request for controller_timelog_edit_after_save hook

2010-06-20 23:33 - Andriy Lesyuk

Status:	Closed	Start date:	2010-06-20
Priority:	Low	Due date:	
Assignee:		% Done:	0%
Category:	Plugin API	Estimated time:	0.00 hour
Target version:			
Description			
The Timelog controller has controller_timelog_edit_before_save hook but does not provide controller_timelog_edit_after_save hook. I believe it should provide the last as well.			
Thanks			

History

#1 - 2010-06-24 06:33 - test test

(snip)

#2 - 2010-08-24 16:49 - Andriy Lesyuk

Guys, please apply this patch! :(

#3 - 2010-08-25 22:12 - Jean-Baptiste Barth

Can you just give us some details about your use case ? As said in other issues, we could add hooks everywhere but it will become hard to maintain. Can't you use rails standard callbacks ? Just tell us a bit more.

#4 - 2010-08-25 22:34 - Andriy Lesyuk

There is an external tool which receives notifications from Redmine about new issues, changes in issues... new time entries, changes in time entries etc. I wrote a very little Redmine plugin responsible for sending notification to this tool.

If I "attach" to standard Rails callbacks the external tool fails to find a corresponding record in database because they do not exist yet at the moment of notification. So I found that using hooks works very well - records are always there already.

#5 - 2010-08-25 22:44 - Jean-Baptiste Barth

Andriy Lesyuk wrote:

If I "attach" to standard Rails callbacks the external tool fails to find a corresponding record in database because they do not exist yet at the moment of notification.

Even with an after_save callback on the right object ? (Issue or TimeEntry I presume)

#6 - 2010-08-25 22:51 - Andriy Lesyuk

Yes. I googled a little for this issue and that's what I've found:

For example, if you invoke an external indexer at this point it won't see the changes in the database.

<http://api.rubyonrails.org/classes/ActiveRecord/Callbacks.html#M001373>

#7 - 2010-08-26 00:27 - Eric Davis

- Category changed from Time tracking to Plugin API

- Assignee set to Eric Davis

Andriy Lesyuk:

Can you post or give me a url to your callback code for this?

The problem with adding hooks in the controller is that they make things really hard for us to change how Redmine works without breaking the hooks.

I'm running into this exact same problem with the Issues controller right now.

#8 - 2010-08-26 00:53 - Jean-Baptiste Barth

- Category changed from Plugin API to Time tracking
- Status changed from New to 7
- Assignee changed from Eric Davis to Jean-Baptiste Barth
- Target version set to 1.0.2

Seems ok to me. Cannot commit this tonight, but I think we'll integrate it in the next version.

#9 - 2010-08-26 07:06 - Jean-Baptiste Barth

- Category changed from Time tracking to Plugin API
- Assignee changed from Jean-Baptiste Barth to Eric Davis

Sorry just saw your reply Eric, didn't want to steal you this issue :)

Is there a place where this kind of problem is described ? Do you have ideas to make these hooks a bit more flexible ? For Issue model the save transaction is wrapped in a specific method with our hooks, can we apply it to other models in your opinion ?

#10 - 2010-08-26 17:44 - Andriy Lesyuk

Not sure if this is what you requested:

```
class IssueHook < Redmine::Hook::ViewListener

  def controller_issues_new_after_save(context = {})
    if context[:issue] && context[:issue].id.present?
      Monkey.instance.tell("+issue:#{context[:issue].id}/cinas")
    end
  end

  def controller_issues_edit_after_save(context = {})
    if context[:time_entry] && context[:time_entry].id.present?
      Monkey.instance.tell("+entry:#{context[:time_entry].id}/cieas")
    end
    if context[:journal] && context[:journal].id.present?
      Monkey.instance.tell("+journal:#{context[:journal].id}/cieas")
    end
  end

  def controller_issues_bulk_edit_after_save(context = { })
    if context[:journal] && context[:journal].id.present?
      Monkey.instance.tell("+journal:#{context[:journal].id}/cibeas")
    end
  end

  def controller_journals_edit_post(context = { })
    if context[:journal] && context[:journal].id.present? && context[:journal].notes.present?
      Monkey.instance.tell("-journal:#{context[:journal].id}/cjep")
    end
  end

  def controller_timelog_edit_after_save(context = { })
    if context[:time_entry] && context[:time_entry].id.present?
      if context[:params][:id].present?
        Monkey.instance.tell("-entry:#{context[:time_entry].id}/ctas")
      else
        Monkey.instance.tell("+entry:#{context[:time_entry].id}/ctas")
      end
    end
  end

  def controller_bulk_time_entries_new_after_save(context = { })
    if context[:time_entry] && context[:time_entry].id.present?
      Monkey.instance.tell("+entry:#{context[:time_entry].id}/cbtenas")
    end
  end

end
```

If not - tell me what do you need and I will post.

#11 - 2010-08-26 23:08 - Eric Davis

Andriy Lesyuk wrote:

Not sure if this is what you requested:

From what I see, you can use observers for Issue, TimeEntry, and Journal using the after_save callback provided by Rails. after_save is fired after the record is created so you should be able to access the id.

#12 - 2010-08-27 00:22 - Andriy Lesyuk

Eric Davis wrote:

Andriy Lesyuk wrote:

Not sure if this is what you requested:

From what I see, you can use observers for Issue, TimeEntry, and Journal using the after_save callback provided by Rails. after_save is fired after the record is created so you should be able to access the id.

This id is sent to the external tool written in Perl. This Perl tool then fetches corresponding records from database. If I use Rails callbacks the tool fails to find the records. So, no, I can't use standard Rails callbacks.

#13 - 2010-08-27 01:56 - Eric Davis

- Target version deleted (1.0.2)

Andriy Lesyuk:

What if you send it to the Perl tool asynchronously or have the Perl script wait for the database to commit the transaction?

If this hook is added, it will end up being removed in a few weeks when I start cleaning up the code in this controller. (See the current IssuesController compared to it 6 months ago to see what kind of cleanup I'm doing)

#14 - 2010-08-27 11:28 - Andriy Lesyuk

Well... I'm not sure that waiting for commit is the best solution in this case. For now I choose patching the code. :) Perhaps in future will try to find some solution not requiring patching.

I checked a little the latest code of IssuesController and found that it still contains hooks. I'm not sure why don't you want to include other hooks like ones suggested by me... It seems that you are just renaming hooks. As soon as you provide a similar after_save hook for timelog I will be happy. :) I don't want you to apply my patch I just want you to provide an after_save hook...

Thanks

#15 - 2010-11-27 00:40 - Eric Davis

- Assignee deleted (Eric Davis)

I am stepping down from working on Redmine. If someone else is interesting in working on this issue, feel free to reassign it to them.

Eric Davis

#16 - 2011-03-23 23:22 - Andriy Lesyuk

- Status changed from 7 to Resolved

I don't need this patch anymore...

#17 - 2013-01-15 22:30 - Jan Niggemann (redmine.org team member)

Closing this, status is resolved since 400 days and more (issue was last updated more than 400 days ago)...

#18 - 2013-01-15 22:37 - Jan Niggemann (redmine.org team member)

- Status changed from Resolved to Closed

Files
