Redmine - Patch #5891

New IssuesController hook for new issues

2010-07-16 00:33 - Nick Peelman

Status:	New	**Start date:**	2010-07-16
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Hook requests	**Estimated time:**	0.00 hour
Target version:			

Description

While working on my CC Addresses plugin, I have come across the need to work with @issue during its creation in order to call @issue.cc_addresses.build before render gets called.

The :controller_issues_new_before_save hook only gets called after the form has been submitted, this ties the hands of plugins trying to extend the issue form (like i'm trying to do, obviously).

Adding the hook below allows you to touch the @issue object before render, letting you build nested_attribute objects, etc.

```
Index: app/controllers/issues_controller.rb
===================================================================
--- app/controllers/issues_controller.rb    (revision 3840)
+++ app/controllers/issues_controller.rb    (working copy)
@@ -145,9 +145,10 @@
     end
     @issue.status = default_status
     @allowed_statuses = ([default_status] + default_status.find_new_statuses_allowed_to(User.curr
ent.roles_for_project(@project), @issue.tracker)).uniq

     if request.get? || request.xhr?
       @issue.start_date ||= Date.today
+      call_hook(:controller_issues_new_prepare, { :params => params, :issue => @issue })
     else
       requested_status = IssueStatus.find_by_id(params[:issue][:status_id]) if params[:issue]
       # Check that the user is allowed to apply the requested status
```

Would be great to see this in the next release, its a relatively minor change. Thanks!

History

#1 - 2010-07-18 19:34 - Nick Peelman

I also found I needed this as well:

```
Index: base.rhtml
===================================================================
--- base.rhtml    (revision 3840)
+++ base.rhtml    (working copy)
@@ -65,6 +65,7 @@

 <div id="footer">
     Powered by <%= link_to Redmine::Info.app_name, Redmine::Info.url %> &copy; 2006-2010 Jean-Philippe Lang
+    <%= call_hook :view_layouts_base_footer_bottom %>
 </div>
 </div>
 <%= call_hook :view_layouts_base_body_bottom %>
```

The current hook at the bottom of the page is nice (though I haven't found a use for it yet) but one at the bottom of the footer like this would make it easier for groups or whoever to add their own copyright or other footer to the bottom, without needing to manually change base.rhtml.

#2 - 2010-07-28 22:38 - Nick Peelman

Due to the changes in 1.0, the diff should look like this:

```
peelman@ubuntu:/var/www/redmine-1.0$ svn diff app/controllers/issues_controller.rb
Index: app/controllers/issues_controller.rb
```

```
===================================================================
--- app/controllers/issues_controller.rb    (revision 3892)
+++ app/controllers/issues_controller.rb    (working copy)
@@ -466,6 +466,7 @@
     @issue.start_date ||= Date.today
     @priorities = IssuePriority.all
     @allowed_statuses = @issue.new_statuses_allowed_to(User.current, true)
+    call_hook(:controller_issues_new_prepare, { :params => params, :issue => @issue })
   end

   def set_flash_from_bulk_issue_save(issues, unsaved_issue_ids)
```

#3 - 2010-08-06 00:42 - Nick Peelman

Nobody?

#4 - 2010-08-06 02:21 - Eric Davis

- Subject changed from New IssuesController hook... to New IssuesController hook for new issues

- Category changed from Issues to Plugin API

Nick Peelman wrote:

> Nobody?

Sorry, there are over 2,000 issues here and less than half a dozen people patrolling the issue list. Sometimes it takes us a bit to get everything.

The hook in the footer is a good one, I'll need to review the one in the controller to see what you need. If you are only adding some objects to new issues, it might be better to hook into Issue's after_initialize that way it's always run.

#5 - 2010-08-06 02:29 - Nick Peelman

Eric Davis wrote:

> Sorry, there are over 2,000 issues here and less than half a dozen people patrolling the issue list. Sometimes it takes us a bit to get everything.
>
> The hook in the footer is a good one, I'll need to review the one in the controller to see what you need. If you are only adding some objects to new issues, it might be better to hook into Issue's after_initialize that way it's always run.

I work on CoRD, or at least i try to. I know the feeling when there's more issues than you have hands. 'Tis one reason I'm working my way up towards helping out more and pitching in with some patches and code here. I do my best to be patient and hack together my own solutions in the mean while :)

You may be right, after_initialize might be a better way. But when you're doing things like I'm doing, that are dependent on modules being enabled and the user having permissions for those modules, how would you go about adding that kind of logic into the issue model/helper? Seems like a better candidate for a controller action, but I'm still learning how to swim in this soup called Ruby...

#6 - 2016-12-30 09:19 - Go MAEDA

- Category changed from Plugin API to Hook requests