Redmine - Defect #5932

hook for :view_repositories_show_contextual is incomplete

2010-07-21 17:30 - Arnaud Martel

Status:	Reopened		**Start date:**	2010-07-21
Priority:	Normal		**Due date:**	
Assignee:			**% Done:**	0%
Category:	Plugin API		**Estimated time:**	0.00 hour
Target version:				
Resolution:			**Affected version:**	

Description

I wrote a plugin to add new commands in repository views (upload files, create directory and delete file/directory) and I noticed that :view_repositories_show_contextual hook lacks of 2 context parameters:

- path
- revision

As a "workaround", I patched the views (show.rhtml, entry.rhtml, changes.rhtml and annotate.rhtml) to replace:

```
<%= call_hook(:view_repositories_show_contextual, { :repository => @repository, :project => @proje
ct }) %>
```

with

```
<%= call_hook(:view_repositories_show_contextual, { :repository => @repository, :project => @proje
ct, :path => @path, :revision => @rev }) %>
```

but it would be better if included in the next release of Redmine...

History

#1 - 2010-07-21 22:20 - Holger Just

As you can see in one of my plugins (line 15), you can get these variables today without having to patch the hook calls.

#2 - 2010-07-21 23:12 - Arnaud Martel

- Status changed from New to Resolved

Many thanks!!! I didn't know how to do that and I really like your solution: I will undo my patches ASAP!!!

#3 - 2010-07-23 00:15 - Holger Just

- Status changed from Resolved to Closed

- Resolution set to Wont fix

#4 - 2010-07-25 03:19 - Jean-Baptiste Barth

- Category set to Plugin API

- Status changed from Closed to Reopened

- Affected version (unused) set to devel

- Resolution deleted (Wont fix)

I disagree... Many things are accessible through context[:controller] but it's not the right way to go for me. Hooks should provide instance variables directly so plugin developers don't have to do hacky things like instance_variable_get. Else, with this argument, we could remove nearly any context variables passed to the hooks ;)

It makes me think of using .send() to bypass object privacy: possible, but not so good in most cases.

Is there a particular reason why we should refuse adding those variables ? Maybe something I missed..