

Redmine - Defect #11151

Functions from helper modules for plugin not accessible

2012-06-13 17:46 - Vitaly Klimov

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Plugin API	Estimated time:	0.00 hour
Target version:		Affected version:	2.0.1
Resolution:	Invalid		

Description

It seems that files under **plugins/plugin_name/helpers** ignored by Rails completely.

To check this, just create some SettingsHelper module in file names **settings_helper.rb** in some plugin folder:

```
module SettingsHelper
  def get_text
    "Plugin description"
  end
end
```

Next, create file in **app/views/settings/** named **_some_plugin_settings.html.erb** with following content:

```
<fieldset class="box tabular">
  <p>
    <%= get_text %>
  </p>
</fieldset>
```

Then add settings property to plugin **init.rb** file:

```
settings(:partial => 'settings/some_plugin_settings')
```

Now, when you try to configure plugin you will get **undefined method (get_text)** error.

History

#1 - 2012-06-14 12:29 - Vitaly Klimov

Update: same applies for files in **controllers** folder as well if they are differ from plugin controllers.

For example, you are not able to add some functions to, sys, **messages_controller.rb** by simply putting file with function in **app/controllers/messages_controller.rb**

#2 - 2012-06-14 13:44 - Jean-Philippe Lang

- Status changed from New to Closed

- Resolution set to Invalid

This is not supposed to work. Your helpers will be loaded automatically for your plugin controllers. But if you want to extend the core helpers, you'll have to get your own modules included by the core helpers, eg.:

```
module MyHelper
  def my_custom_method
    # do something
  end
end

require 'settings_helper'
SettingsHelper.send :include, MyHelper
```

#3 - 2012-06-14 14:28 - Vitaly Klimov

Thanks. I know about this way of extending helpers. But method described by me above was perfectly normal while plugins were in **vendor/plugins**

folder. I guess there should be some note about it somewhere in docs.