

Redmine - Patch #14905

Add info to documentation: RedmineUpgrade page

2013-09-14 13:02 - Sergei Babski

| | | | |
|------------------------|---------------|------------------------|-----------|
| Status: | Closed | Start date: | |
| Priority: | Normal | Due date: | |
| Assignee: | | % Done: | 0% |
| Category: | Documentation | Estimated time: | 0.00 hour |
| Target version: | | | |

Description

After upgrading to redmine 2.3.3 on Rails 3.2.13 I've got an 500 error on issues page.
log

```
ActionView::Template::Error (undefined method `remote_function' for #<#<Class:0xb5aaea00>:0xb5aacf5c>):  
33: </div>  
34:  
35: <p><%= submit_tag l(:button_save) >  
36: <= link_to_remote l(:label_preview),  
37: { :url => { :controller => 'wiki', :action => 'preview',  
38: :project_id => @project, :id => @page.title },  
39: :method => :post,  
lib/plugins/prototype_legacy_helper/lib/prototype_legacy_helper.rb:178:in `link_to_remote'  
app/views/wiki/edit.html.erb:36:in `_app_views_wiki_edit_html_erb__89008286__623550738'  
app/views/wiki/edit.html.erb:5:in `_app_views_wiki_edit_html_erb__89008286__623550738'  
app/controllers/wiki_controller.rb:62:in `show'
```

I was able to fix it by adding following code to lib/plugins/prototype_legacy_helper/lib/prototype_legacy_helper.rb

```
CALLBACKS = Set.new([ :create, :uninitialized, :loading, :loaded,  
:interactive, :complete, :failure, :success ] +  
(100..599).to_a)  
  
def build_callbacks(options)  
  callbacks = {}  
  options.each do |callback, code|  
    if CALLBACKS.include?(callback)  
      name = 'on' + callback.to_s.capitalize  
      callbacks[name] = "function(request){#{code}}"  
    end  
  end  
  callbacks  
end  
  
def options_for_ajax(options)  
  js_options = build_callbacks(options)  
  
  js_options['asynchronous'] = options[:type] != :synchronous  
  js_options['method'] = method_option_to_s(options[:method]) if options[:method]  
  js_options['insertion'] = "'#{options[:position].to_s.downcase}'" if options[:position]  
  js_options['evalScripts'] = options[:script].nil? || options[:script]  
  
  if options[:form]  
    js_options['parameters'] = 'Form.serialize(this)'  
  elsif options[:submit]  
    js_options['parameters'] = "Form.serialize('#{options[:submit]}')"  
  elsif options[:with]  
    js_options['parameters'] = options[:with]  
  end  
end
```

```

if protect_against_forgery? && !options[:form]
  if js_options['parameters']
    js_options['parameters'] << " + '&"
  else
    js_options['parameters'] = ""
  end
  js_options['parameters'] << "#{request_forgery_protection_token}=" + encodeURIComponent("#{escape_javascript form_authenticity_token}")"
end

options_for_javascript(js_options)
end

def remote_function(options)
  javascript_options = options_for_ajax(options)

  update = ''
  if options[:update] && options[:update].is_a?(Hash)
    update = []
    update << "success:'#{options[:update][:success]}'" if options[:update][:success]
    update << "failure:'#{options[:update][:failure]}'" if options[:update][:failure]
    update = '{' + update.join(',') + '}'
  elsif options[:update]
    update << "'#{options[:update]}'"
  end

  function = update.empty? ?
    "new Ajax.Request(" :
    "new Ajax.Updater(#{update}, "

  url_options = options[:url]
  function << "'#{html_escape(escape_javascript(url_for(url_options)))}'"
  function << ", #{javascript_options}"

  function = "#{options[:before]}; #{function}" if options[:before]
  function = "#{function}; #{options[:after]}" if options[:after]
  function = "if (#{options[:condition]}) { #{function}; }" if options[:condition]
  function = "if (confirm('#{escape_javascript(options[:confirm])}')) { #{function}; }" if options[:confirm]

  return function.html_safe
end

```

from here <http://rishabh23.blogspot.com/2012/11/prototypelegacyhelper-for-rails-309.html>

It would be nice to include this info on [RedmineUpgrade](#) page of the Documentation.

History

#1 - 2013-09-14 14:31 - Jean-Philippe Lang

- Status changed from New to Closed

There's no prototype_legacy_helper plugin in Redmine 2.3.3. Try to use a fresh code base instead, thanks.