

Redmine - Feature #9759

HTML minification and compression

2011-12-08 21:55 - luigifab !

| | | | |
|--|--------------------------|------------------------|-------------------|
| Status: | New | Start date: | |
| Priority: | Normal | Due date: | |
| Assignee: | | % Done: | 0% |
| Category: | Code cleanup/refactoring | Estimated time: | 0.00 hour |
| Target version: | | | |
| Resolution: | | | |
| Description | | | |
| To improve page loading, you can use Tidy for compressing HTML code (show source code of this example). | | | |
| Yes the gain will be low, but this is a good start before to minify CSS and JavaScript files. | | | |
| To further improve HTML compression, you can also gzip output. Sorry, I don't know RUBY, but in PHP, you can gzip files with output buffering . | | | |
| Related issues: | | | |
| Related to Redmine - Feature #4796: Rails 3 support | | Closed | 2010-02-10 |

History

#1 - 2011-12-09 09:11 - Etienne Massip

Rails 3.1 with sprockets enhance greatly serving of assets.

#2 - 2013-01-12 20:06 - Daniel Felix

Well any news for this ticket?

It relates to Rails 3 support which is implemented. Currently redmine serves quite good and small source.

#3 - 2015-04-30 21:52 - luigifab !

Hi all, I have done a beautiful code to GZIP (because Thin+lighttpd can't do this) and minify (with Tidy) HTML page!

In the `application_controller.rb`:

```
require 'base64'
...
class ApplicationController < ActionController::Base
  ...
  after_filter :tidy
  def tidy
    if response.body =~ /<html/i
      response.header['Content-Encoding'] = 'gzip'
      response.body = `php ../../myfile.php #{Base64.encode64(response.body).gsub(/\n/, ' ')} `
    end
  end
end
```

And my PHP file:

```
error_reporting(E_ALL);
ini_set('display_errors', 1);

$html = base64_decode(str_replace('ø', "\n", $argv[1]));
$html = str_replace('></option>', '>&nbsp;</option>', $html);
$html = str_replace('></button>', '>&nbsp;</button>', $html);
$html = myTidyFunction($html); // my custom function that call Tidy
$html = str_replace('>&nbsp;</option>', '></option>', $html);
$html = str_replace('>&nbsp;</button>', '></button>', $html);
$html = str_replace('<meta charset="utf-8"/>',
'<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>', $html);
$html = gzencode($html, 9, FORCE_GZIP);

echo $html;
```

Yeah, it's horrible!
But it works so very good.