

## Redmine - Patch #7204

### Caching of public pages, requested from anonymous users (10x - 20x speed improvements)

2010-12-31 03:47 - Massimo Zaniboni

<b>Status:</b>	New	<b>Start date:</b>	2010-12-31
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>			

#### Description

I need a quicker access for public pages, requested from anonymous users.

I can not use a caching reverse proxy because the web proxy does not know when a page is requested from a registered user, or when the page is requested from an anonymous user.

In this patch the Ruby/Redmine application controller returns a page from the cache with ~ 10x - 20x speedup respecting normal pages. The caching is applied only if the user is anonymous (not logged) and only if the page does not contains forms with an authenticity token.

The cached page is different according the locale of user, so different pages for different locales are returned correctly.

The Redmine administrator can configure the cache refresh rate using the main Settings form.

The caching is performed using the caching method specified in the config/additional\_environment.rb file. I added something like

```
config.action_controller.perform_caching = true
config.action_controller.cache_store = :file_store, File.join(File.dirname(__FILE__), '../tmp/cache')
```

I tested the patch on version 1.0.5, but it is not yet on a server in production.

The patch can be applied to r4595 or Git commit:6659c630b264325

Warnings / sorry in advance :-)

- this is my first ruby code
- this is my first ruby on rails code
- I'm not a native english speaker, and probably my labels/texts is not completely correct

Many thanks for the good product that is Redmine!

#### History

##### #1 - 2010-12-31 09:23 - Felix Schäfer

At least that setting:

```
config.action_controller.perform_caching = true
```

Is already part of the production environment. What environment and rails server do you use, and what kind of load are you having/expecting that this is a problem for you?

##### #2 - 2010-12-31 13:06 - Massimo Zaniboni

Regarding caching, I had the same problem described in <http://www.redmine.org/boards/2/topics/2432>

I'm using nginx + phusion passenger + enterprise ruby on a local machine, and I have no public server up to date. I don't know in advance what will be my load.

### #3 - 2010-12-31 13:27 - Felix Schäfer

Massimo Zaniboni wrote:

*Regarding caching, I had the same problem described in <http://www.redmine.org/boards/2/topics/2432>*

That's a wee bit old...

*I don't know in advance what will be my load.*

Then you're probably trying to solve a problem you don't have.

Anyway, my experience is that the rails-provided facilities for caching are good enough (if configured correctly, of course), we don't get that much traffic over at <https://orga.fachschafte.org/>, but the install is quite "snappy" nonetheless :-)

### #4 - 2010-12-31 15:37 - Massimo Zaniboni

I didn't find a lot of info about proper configuration of Redmine caching on the web.

I performed some test, without effective results. Then I studied Rails caching support, but I didn't find related instructions inside Redmine code, and my impression (but I'm not an expert), it is that Redmine uses very few Rails caching features.

### #5 - 2010-12-31 16:17 - Jean-Philippe Lang

Redmine uses few caching technics indeed, only where it can improve response times significantly: activity rendering, diff rendering, textile rendering. This is generally not so easy to implement and we don't have much issues about performance reported here.

Concerning your patch, it looks like a good start and could be easily implemented as a plugin. I have a few comments:

- it seems reasonable to cache GET requests only
- how do you deal with parameters? Your cache key uses the request path only.

### #6 - 2011-01-02 14:53 - Massimo Zaniboni

*Concerning your patch, it looks like a good start and could be easily implemented as a plugin. I have a few comments: - it seems reasonable to cache GET requests only*  
*- how do you deal with parameters? Your cache key uses the request path only.*

Good questions: add to the list that I don't know HTTP :-)

I focused better my problem: if Redmine pages will be cacheable, then http requests can be answered directly from a web proxy cache, that is 100 ~ 1000 faster than Redmine.

I will try to create a plugin, that when enabled: - allows safe caching of pages for anonymous users, setting properly cache related http fields, removing unused cookies, and so on;

- provide default pages for Issues, Roadmap and Activity pages, accessed from anonymous users, without customizable forms, but correctly cacheable;

In this way I can use Redmine both for project development (few registered users), and project presentation (hopefully many visitors), without any concerns about speed.

Is it a naive approach? Is it an already solved problem? Thanks in advance for any positive/negative feedback.

**#7 - 2011-01-02 17:17 - Felix Schäfer**

Rails already does a lot of caching in production: page caching, fragment caching, and so on. I'd suggest you read up on those topics and look for the common or best practices regarding caching in rails before trying to reinvent the wheel.

**#8 - 2011-01-02 17:25 - Felix Schäfer**

Massimo Zaniboni wrote:

| *[...] and project presentation (hopefully many visitors), without any concerns about speed.*

Oh, and regarding that, I can only repeat myself: don't try to solve a problem you haven't run into or even approached in any significant way.

If you have numbers about expected traffic, do some tests with these numbers, try to find the bottlenecks and fix those, what you are trying to do in many ways replicates existing functionality (e.g. reverse proxies). If your only gripe with redmine is that the reverse proxy has no way to know if the user is anonymous, find a way to tell the proxy instead of trying to replicate the proxy in redmine.

Anyway, my 0,02€ on that topic.

**#9 - 2011-01-02 18:24 - Massimo Zaniboni**

| *Oh, and regarding that, I can only repeat myself: don't try to solve a problem you haven't run into or even approached in any significant way.*

From a business point of view, you are completely right. But you should also consider that in the meantime I learn something about ruby on rails, web-proxy, and maybe I can release some work useful also to other people.

| *If your only gripe with redmine is that the reverse proxy has no way to know if the user is anonymous, find a way to tell the proxy instead of trying to replicate the proxy in redmine.*

Exactly the approach I described in my past message...

**#10 - 2011-01-02 18:55 - Felix Schäfer**

Massimo Zaniboni wrote:

| *If your only gripe with redmine is that the reverse proxy has no way to know if the user is anonymous, find a way to tell the proxy instead of trying to replicate the proxy in redmine.*

| Exactly the approach I described in my past message...

You're right, sorry, I was still thinking of your first approach.

**#11 - 2011-01-10 00:30 - Massimo Zaniboni**

- File *redmine.patch* added

I created a plugin, for caching web-pages, using Varnish web cache, that completely substitute/improve my previous patch.

[https://github.com/massimo-zaniboni/Redmine\\_cacheable\\_pages](https://github.com/massimo-zaniboni/Redmine_cacheable_pages)

It is not already tested in production.

This plugin works only applying the attached patch to Redmine. Maybe there is a better method for writing it, without modifying Redmine.

The Redmine forms will contain the `authenticity_code`, but it is a code associated to a public session exchanged between Varnish reverse web-proxy and Redmine server, and only for anonymous/not-logged users. I don't think this poses security problems. I were not able to remove sessions and authenticity code from Rails/Redmine.

I changed from POST to GET method inside ISSUES, when APPLY a query, in order to enable caching of ISSUE pages.

**#12 - 2011-01-10 00:49 - Massimo Zaniboni**

Some clarifications, in order to facilitate inspecting of the code.

The logic of the plugin is inside

[https://github.com/massimo-zaniboni/Redmine\\_cacheable\\_pages/blob/master/lib/redmine\\_cacheable\\_pages/hooks/cacheable\\_pages\\_hooks.rb](https://github.com/massimo-zaniboni/Redmine_cacheable_pages/blob/master/lib/redmine_cacheable_pages/hooks/cacheable_pages_hooks.rb)

and

[https://github.com/massimo-zaniboni/Redmine\\_cacheable\\_pages/blob/master/varnish/config\\_example.vcl](https://github.com/massimo-zaniboni/Redmine_cacheable_pages/blob/master/varnish/config_example.vcl)

Because I were not able to remove Redmine session cookies from Redmine code (they are lazily produced whenever there is an access to a session field), I removed it on the Varnish side. It is not very elegant. Nginx configuration language was not powerful enough for coding all these transformations, so I used Varnish.

**Files**

---

cache.patch	4.82 KB	2010-12-31	Massimo Zaniboni
redmine.patch	2.54 KB	2011-01-09	Massimo Zaniboni