

Redmine - Feature #11445

Switch from Prototype to JQuery

2012-07-19 06:37 - Jean-Philippe Lang

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Jean-Philippe Lang	% Done:	0%
Category:	Code cleanup/refactoring	Estimated time:	0.00 hour
Target version:	2.1.0		
Resolution:	Fixed		
Description			

Associated revisions

Revision 10068 - 2012-07-22 15:29 - Jean-Philippe Lang

JQuery in, Prototype/Scriptaculous out (#11445).

Revision 10069 - 2012-07-22 16:43 - Jean-Philippe Lang

Use JQuery Datepicker (#11445).

Revision 10070 - 2012-07-22 17:42 - Jean-Philippe Lang

Use JQuery Dialog (#11445).

History

#1 - 2012-07-21 20:53 - MaximAL

+1

#2 - 2012-07-21 22:34 - Travis Warlick

Has this process been started? If so, where is the code, and may I assist? (I would also like to create a plugin to back port the changes 2.0.x if it's possible -- although I haven't thought of all the ramifications to using such a plugin, so any advise would be appreciated.)

#3 - 2012-07-21 23:22 - Jean-Philippe Lang

Travis Warlick wrote:

Has this process been started? If so, where is the code, and may I assist?

Sure it has started and should be committed in the next few days. I'm no JQuery expert, so any patch to clean up the code will be appreciated.

#4 - 2012-07-22 14:33 - Etienne Massip

Isn't it a good occasion to activate the Rails asset pipeline (but with compile option activated at production level so that there is no need to precompile the assets at deployment time) and make use of [CoffeeScript](#) parser with jquery?

Also related to this asset pipeline thing, for the time being the prototype-rails gem is required but the prototype javascript files are also included in the public folder, which is redundant or there is not point to require prototype-rails gem.

#5 - 2012-07-22 15:59 - Etienne Massip

Or even add js integration testing using [Selenium](#)?

#6 - 2012-07-26 19:08 - Jean-Philippe Lang

- Status changed from New to Closed

- Resolution set to Fixed

Etienne Massip wrote:

Isn't it a good occasion to activate the Rails asset pipeline (but with compile option activated at production level so that there is no need to precompile the assets at deployment time)

I wouldn't be in favour of compilation at runtime. And the Rails guide says: *"Live Compilation: This mode uses more memory, performs more poorly than the default and is not recommended"*.

and make use of [CoffeeScript](#) parser with jQuery?

I have never used CoffeeScript personally, so maybe later.

Also related to this asset pipeline thing, for the time being the prototype-rails gem is required but the prototype javascript files are also included in the public folder, which is redundant or there is not point to require prototype-rails gem.

The prototype-rails gem is gone.

#7 - 2012-07-29 23:26 - Etienne Massip

Also note that the `addFormObserversForDoubleSubmit` function can be replaced by the use of `data-disable-with` HTML5 attribute supported by the [jQuery ujs gem](#).

#8 - 2012-07-30 19:20 - Etienne Massip

Jean-Philippe Lang wrote:

Etienne Massip wrote:

Isn't it a good occasion to activate the Rails asset pipeline (but with compile option activated at production level so that there is no need to precompile the assets at deployment time)

I wouldn't be in favour of compilation at runtime. And the Rails guide says: *"Live Compilation: This mode uses more memory, performs more poorly than the default and is not recommended"*.

I don't know, they also say : On the first request the assets are compiled and cached as outlined in development above, and the manifest names used in the helpers are altered to include the MD5 hash..

What I thought was to advise admin to precompile assets at deployment time but keep the compile parameter on to degrade (not much) gracefully if the assets aren't precompiled (if the parameter is off then the application displays an error page to the end user).

If the compilation is run at each request it's obviously not a good idea, but if it is run only at first time it might be worth it?

and make use of [CoffeeScript](#) parser with jQuery?

I have never used CoffeeScript personally, so maybe later.

Feels like Ruby, could be better to do it earlier than later (but not than never) because of the amount of work?

#9 - 2012-07-31 15:07 - Jean-Baptiste Barth

Etienne Massip wrote:

What I thought was to advise admin to precompile assets at deployment time but keep the compile parameter on to degrade (not much) gracefully if the assets aren't precompiled (if the parameter is off then the application displays an error page to the end user).

If the compilation is run at each request it's obviously not a good idea, but if it is run only at first time it might be worth it?

I'm personally OK with that, but I think we must think of people with no much technical skills, it can really be a pain to make the asset pipeline work. There are 3 major reasons imho:

1. you need a JS runtime either on your development (deploy?) machine or your production server. This is not trivial to achieve on Windows for instance
2. your filesystem might be read-only in production, or maybe some subdirectories might be read-only for the user who runs the application server, so it might lead to weird issues that are hard to understand for lambda users
3. fingerprints in assets can be hard to make work (even for "us", developers)

Btw, maybe we can consider the following approach :

- asset pipeline could be used "internally" in the development process but turned off in stable releases
- precompilation would take place when the release manager builds the tarball ; this way he could build a one-file application.js and application.css
- a precompilation task would be added to the redmine:plugins tasks, with graceful fallback if no JS runtime is present (in this case, stop the precompilation, and keep copying plugins' assets individually)

I think the asset pipeline would actually be a great way to ease the management of Redmine instances with many-many plugins (my typical instance has 15+ plugins, and twice this amount of custom JS/CSS, but I'm not a typical user I think).

Jean-Philippe: if it's a total no-no, fine, let's stick with the actual behaviour. If you want us to test this approach, maybe Etienne and I could work on such a feature, write some docs about it and consider edge cases, and then come back to you with a proposal in a dedicated ticket.

#10 - 2012-07-31 18:29 - Jean-Philippe Lang

It's not a no for me and it would be nice to see how plugins can take advantage of it in the real world.

But Redmine is not easy to install especially for those who are not familiar with rails or even with deploying a web application, and I don't want to make it harder. Letting end users compile javascripts is not an option. It can be precompiled so it should be precompiled if we ever use the asset pipeline. In release tarballs and maybe in the repository too. And yes, we should discuss this in a specific ticket since it's not related to JQuery.

#11 - 2012-07-31 18:56 - Etienne Massip

Jean-Baptiste Barth wrote:

Etienne Massip wrote:

What I thought was to advise admin to precompile assets at deployment time but keep the compile parameter on to degrade (not much) gracefully if the assets aren't precompiled (if the parameter is off then the application displays an error page to the end user).

If the compilation is run at each request it's obviously not a good idea, but if it is run only at first time it might be worth it?

I'm personally OK with that, but I think we must think of people with no much technical skills, it can really be a pain to make the asset pipeline work.

The suggestion was made in this direction : "serious" administrators will want to precompile the assets before the first deployment so that there will be no "first request" precompilation and "casual" admins will deploy the instance assets as they are straight from the box so that the compile setting set to "on" will prevent any server error due to the missing precompiled assets.

Still if this setting does not force precompiling of assets at each request.

There are 3 major reasons imho:

1. you need a JS runtime either on your development (deploy?) machine or your production server. This is not trivial to achieve on Windows for instance

Do you really? There are gems needed to uglify javascript and to compile SASS to CSS and CS to JS but is there any need for a JS runtime ?

1. your filesystem might be read-only in production, or maybe some subdirectories might be read-only for the user who runs the application server, so it might lead to weird issues that are hard to understand for lambda users

Same for plugin assets today and

1. fingerprints in assets can be hard to make work (even for "us", developers)

They are computed automatically, aren't they?

Btw, maybe we can consider the following approach :

- asset pipeline could be used "internally" in the development process but turned off in stable releases
- precompilation would take place when the release manager builds the tarball ; this way he could build a one-file application.js and application.css

I was wondering if this is the workflow RoR engineers had in mind when they implemented the pipeline, actually.

- a precompilation task would be added to the redmine:plugins tasks, with graceful fallback if no JS runtime is present (in this case, stop the precompilation, and keep copying plugins' assets individually)

I think the asset pipeline would actually be a great way to ease the management of Redmine instances with many-many plugins (my typical instance has 15+ plugins, and twice this amount of custom JS/CSS, but I'm not a typical user I think).

RM differs from RoR main path here because RoR 3.x plugins are supposed to be bundled as gems and use their own engine pipeline (but can use resources of the main app pipeline). I don't know if handling the plugin assets together with the main app ones would be an easy task in RM and I fear it wouldn't.

#12 - 2012-08-12 13:27 - Terence Mill

Does it mean that all ajax calls using jquery now?

#13 - 2012-08-13 10:23 - Jean-Philippe Lang

Terence Mill wrote:

Does it mean that all ajax calls using jquery now?

Yes.