

Redmine - Patch #37664

Preload all stylesheets and JavaScripts

2022-09-11 23:37 - Takashi Kato

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Performance	Estimated time:	0.00 hour
Target version:	Candidate for next major release		
Description			
All browsers usually block HTML from rendering when loading stylesheets and javascript. Preloading the stylesheet and javascript with the link element or Link header can avoid this block.			
Rails 6.1 has a feature that automatically preloads stylesheets and javascript, so enable it.			
Please see https://github.com/rails/rails/pull/39939 and https://github.com/rails/rails/pull/40882 .			

History

#1 - 2022-09-13 10:46 - Go MAEDA

I think the patch is expected to add rel="preload" to link tags, right?

I tried out the patch but found no change in link tags.

```
<link rel="stylesheet" media="all" href="/stylesheets/jquery/jquery-ui-1.13.1.css?1655625104" />
<link rel="stylesheet" media="all" href="/stylesheets/tribute-5.1.3.css?1589811489" />
<link rel="stylesheet" media="all" href="/stylesheets/application.css?1660262973" />
<link rel="stylesheet" media="all" href="/stylesheets/responsive.css?1655625104" />
```

#2 - 2022-09-13 12:22 - Takashi Kato

This patch adds a [Link header](#) to the HTTP response header as follows. The effect is the same as adding a link element to html.

```
Link: </stylesheets/jquery/jquery-ui-1.13.1.css?1662818440>; rel=preload; as=style; nopush,
</stylesheets/tribute-5.1.3.css?1662818440>; rel=preload; as=style; nopush,
</stylesheets/application.css?1663018932>; rel=preload; as=style; nopush,
</stylesheets/responsive.css?1662818440>; rel=preload; as=style; nopush,
</javascripts/jquery-3.6.0-ui-1.13.1-ujs-6.1.3.1.js?1662818440>; rel=preload; as=script; nopush,
</javascripts/tribute-5.1.3.min.js?1662818440>; rel=preload; as=script; nopush,
</javascripts/tablesort-5.2.1.min.js?1662818440>; rel=preload; as=script; nopush,
</javascripts/tablesort-5.2.1.number.min.js?1662818440>; rel=preload; as=script; nopush,
</javascripts/application.js?1662818440>; rel=preload; as=script; nopush,
</javascripts/responsive.js?1662818440>; rel=preload; as=script; nopush
```

#3 - 2022-09-13 14:13 - Go MAEDA

- Target version set to Candidate for next major release

#4 - 2022-09-25 05:24 - Go MAEDA

- Target version changed from Candidate for next major release to 5.1.0

Setting the target version to 5.1.0.

#5 - 2022-09-27 01:41 - Mischa The Evil

I am not able to test this myself, but I wonder: does this 'play nicely' with Redmine's custom implementations of stylesheet_link_tag (source:trunk/app/helpers/application_helper.rb@21848#L1664) and javascript_include_tag (source:trunk/app/helpers/application_helper.rb@21848#L1698)? I.e. does it work for third-party plugin resources too?

#6 - 2022-09-27 04:47 - Go MAEDA

- Target version changed from 5.1.0 to Candidate for next major release

Mischa The Evil wrote:

I am not able to test this myself, but I wonder: does this 'play nicely' with Redmine's custom implementations of stylesheet_link_tag ([source:trunk/app/helpers/application_helper.rb@21848#L1664](https://source.trunk/app/helpers/application_helper.rb@21848#L1664)) and javascript_include_tag ([source:trunk/app/helpers/application_helper.rb@21848#L1698](https://source.trunk/app/helpers/application_helper.rb@21848#L1698))? I.e. does it work for third-party plugin resources too?

I found the patch properly handles third-party theme's stylesheet and JavaScript but does not for plugin assets.

#7 - 2022-10-06 01:05 - Takashi Kato

- File 0002-Add-integration-test.patch added

Preloading doesn't work for plugin assets because calling from call_hook lost the context, not the helper customization.

Using the context when calling the hook works as expected.

Files

0001-Preload-all-stylesheets-and-JavaScripts.patch	848 Bytes	2022-09-11	Takashi Kato
0002-Add-integration-test.patch	2.99 KB	2022-10-05	Takashi Kato