

Redmine - Defect #40528

Hangs when requesting after code changes in development mode

2024-04-05 09:35 - Katsuya HIDAHA

Status:	Confirmed	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		Affected version:	
Resolution:			

Description

When running the rails server in development mode, it may hang when requesting after code changes.

Reproduction Steps

1. Check out trunk ([r22777](#)) and start the rails server in development mode.
2. Log in to Redmine.
3. Make changes to some code and reload the Redmine page.
 - Reproduction may occur simply by adding comments or executing the touch command.

Expected Behavior

The request succeeds, and the page transition is successful.

Actual Behavior

Requests frequently hang, and the page transition does not occur, remaining in a loading state.

The rails server logs in this state are as follows:

```
Started GET "/issues" for ::1 at 2024-04-05 10:59:49 +0900
```

In this state, it is not possible to stop the server with Ctl+C (SIGTERM). It is necessary to stop it each time with SIGKILL (kill -9).

Environment

Environment:

```
Redmine version      5.1.2.devel
Ruby version         3.3.0-p0 (2023-12-25) [aarch64-linux]
Rails version        7.1.2
Environment          development
Database adapter     SQLite
Mailer queue         ActiveSupport::QueueAdapters::AsyncAdapter
Mailer delivery      smtp
```

Redmine settings:

```
Redmine theme       Default
```

SCM:

```
Subversion          1.14.2
Mercurial            6.3.2
Cvs                  1.12.13
Bazaar               3.3.2
Git                  2.39.2
Filesystem
```

Redmine plugins:

```
no plugin installed
```

The issue was confirmed on the following operating systems:

- macOS 14.4.1 (Apple M2)
- Ubuntu 23.10

Investigation

- This issue also reproduces with Ruby 3.3.0.
- The issue might have started occurring after the update to Rails 7.1 ([r22488](#)). At least in my environment, it hasn't reproduced so far with the previous revision [r22487](#).
- The issue does not reproduce when starting Puma with a single thread (the default thread count is 5):

```
PUMA_MIN_THREADS=1 PUMA_MAX_THREADS=1 bin/rails s
```

- Referring to the contents of the [Puma project's issue](#), I conducted an investigation using the `ActionDispatch::DebugLocks` middleware. As a result, I observed logs similar to [this comment](#).

History

#1 - 2024-04-05 09:50 - Katsuya HIDAKA

Investigation

This issue also reproduces with Ruby 3.3.0.

Sorry, the above "Ruby 3.3.0" is incorrect. The correct version is "Ruby 3.2.3."

#2 - 2024-04-07 06:09 - Go MAEDA

- Status changed from New to Confirmed