

[Waiting for dev...](#)

Ruby, Vim, Unix...

- [RSS](#)

<input type="text" value="Search"/>
<input type="text" value="Navigate... ▼"/>

- [Blog](#)
- [Archives](#)
- [About Me](#)

Stop and Restart Easily a Rails Server

May 3rd, 2014 11:09 am

I like to start rails server as a daemon with `rails server -d`, because I don't want a terminal window or a tmux pane just hanging around with the server foreground process.

But each time a new gem is added to the application or some changes are made in some configuration file, it is a pain to manually look for the pid in `tmp/pids/server.pid`, kill it and start again the server.

I wished if I could have convenient `rails start`, `rails stop` and `rails restart` commands to help me in the workflow. And it is very easy to get.

Bellow you'll find a simple bash (or zsh) function that should be added to your `.bashrc` (or `.zshrc`). Once done, you'll be able to do things like the following:

```
1 rails start # To start the server in development environment
2 rails start production # To start the server in production environment
3 rails stop # To stop the server
4 rails stop -9 # To stop the server sending -9 kill signal
5 rails restart # To restart the server in development environment
6 rails restart production # To restart the server in production environment
7 rails whatever # Will send the call to original rails command
```

Here it is the function:

```
1 function rails() {
2   if [ "$1" = "start" ]; then
3     if [ "$2" = "" ]; then
4       RENV="development"
5     else
6       RENV="$2"
7     fi
```

```

8 rails server -d -e "$RENV"
9 return 0
10 elif [ "$1" = "stop" ]; then
11     if [ -f tmp/pids/server.pid ]; then
12         kill $2 $(cat tmp/pids/server.pid)
13         return 0
14     else
15         echo "It seems there is no server running or you are not in a rails project root directory"
16         return 1
17     fi
18 elif [ "$1" = "restart" ]; then
19     rails stop && rails start $2
20 else
21     command rails $@
22 fi;
23 }

```

rails_server_helper.bash hosted with ❤ by GitHub

[view raw](#)

You see, the code has not too much mystery. Inside the `else` block, `command` is a bash built-in command which executes the given command (in this case `rails`), ignoring any shell function named as it (our function). So, we use it to default to original rails executable when the first argument is not `start`, `stop` or `restart`, passing any other argument with `#@`.

If you don't feel right overwriting `rails`, just change the function name and remove the `else` block.

Posted by Marc Busqué May 3rd, 2014 11:09 am

[Tweet](#)

[Distributable and organized dotfiles with homeshick and mr »](#)

Comments

9 Comments

Waiting for dev

[1](#) Login ▾

♥ Recommend 1 [Share](#)

Sort by Best ▾



Join the discussion...



Gerard Gold • a year ago

Quick & dirty -- works on Kali Linux (Debian) --
<https://github.com/GGTechGuru/...>

^ | ▾ • Reply • Share ›

— | 🚩



Shaun Stanislaus • 2 years ago

— | 🚩



i try pasting your this bash commands into my bashrc it doesn't work.

could you kindly write the full functions to perform

the following please:

rails start # To start the server in development environment

rails start production # To start the server in production environment

rails stop # To stop the server

rails stop -9 # To stop the server sending -9 kill signal

rails restart # To restart the server in development environment

rails restart production # To restart the server in production environment

rails whatever # Will send the call to original rails command

thank you!

^ | v • Reply • Share ›



Shaun Stanislaus → Shaun Stanislaus • 2 years ago

shaunstanislaus@Master:~/Desktop/ruby_tutorial/blog\$ rails whatever

Error: Command 'whatever' not recognized

Usage: rails COMMAND [ARGS]

The most common rails commands are:

generate Generate new code (short-cut alias: "g")

console Start the Rails console (short-cut alias: "c")

server Start the Rails server (short-cut alias: "s")

dbconsole Start a console for the database specified in config/database.yml
(short-cut alias: "db")

new Create a new Rails application. "rails new my_app" creates a
new application called MyApp in "./my_app"

In addition to those, there are:

application Generate the Rails application code

destroy Undo code generated with "generate" (short-cut alias: "d")

plugin new Generates skeleton for developing a Rails plugin

runner Run a piece of code in the application environment (short-cut alias: "r")

All commands can be run with -h (or --help) for more information.

shaunstanislaus@Master:~/Desktop/ruby_tutorial/blog\$

^ | v • Reply • Share ›



Marc Busqué Mod → Shaun Stanislaus • 2 years ago

Maybe you tried it before I changed a bug in the line with the `command` command?

It must be;

command rails \$@

instead of:

command rails #@

Hope it helps. If not, please tell me exactly what it is not working.

^ | v • Reply • Share ›



Shaun Stanislaus → Marc Busqué • 2 years ago

Okay, it worked. you are right about the \$@ and #@

^ | v • Reply • Share ›



Marc Busqué Mod → Shaun Stanislaus • 2 years ago

Ok, great!

^ | v • Reply • Share ›



Shaun Stanislaus → Marc Busqué • 2 years ago

Hi Marc, i don't really know bash commands well and i am new to ruby, but i do know the long convention form command.

i pasted the bash scripts you created for making rails commands easier into my ~/.bashrc and source ~/.bashrc but it is not working.

i did rails start, it does not recognise the command.

could you edit the commands above and let me know?

i don't see the word "production" inside your bash scripts you created aliasing the production environment command, you didn't create it as well as the word "whatever".

^ | v • Reply • Share ›



Marc Busqué Mod → Shaun Stanislaus • 2 years ago

"whatever" is just a placeholder. For example "generate" in "generate". Maybe I should make it clear...

The environment is passed as an argument, that's \$2, and it can be 'production' or any other. If it is not set, it defaults to 'development'

^ | v • Reply • Share ›



Shaun Stanislaus → Marc Busqué • 2 years ago

thank you!

^ | v • Reply • Share ›

ALSO ON WAITING FOR DEV

Rails: Rescue From All Foreign Key Violations at Once

9 comments • 2 years ago •

Ruby Gems Could Harm Your Memory

8 comments • 22 days ago •



Marc Busqué — Thanks, good initiative! I updated



Robert Fletcher — Yeah, I could see it being okay for a pretty basic CRUD app, but things rarely stay simple for long. We do have one use case



the article with the link.

✉ Subscribe Add Disqus to your site Add Disqus Add Privacy

Recent Posts

- [Ruby Gems Could Harm Your Memory](#)
- [Rails: Rescue From All Foreign Key Violations at Once](#)
- [Read Your Newsbeuter Feeds in Epub Format](#)
- [Append Issue Number to Commit Message Automatically With Git Hooks](#)
- [Distributable and Organized Dotfiles With Homeshick and Mr](#)

GitHub Repos

- [castle-vim](#)
vim dotfiles
- [castle-zsh](#)
My zsh castle
- [waiting-for-dev.github.io](#)
Personal weblog
- [elixir_state](#)
Trying to understand how state is managed in a functional language like Elixir
- [system_bootstrap](#)
My system bootstrap
- [string-direction](#)
Automatic detection of text direction (ltr, rtl or bidi) for strings in Ruby
- [vim-www](#)
Toolbox to open & search URLs from vim
- [zass](#)
Sass grid framework 100% fluid, semantic, clean and allowing infinite nesting level with the use of one single mixin.
- [castle-tmux](#)
tmux dotfiles
- [landing_page](#)

Minimal, flexible, unobstrusive rails 4 and 3.2 engine for an email subscription form. It collects name, email and locale used. Optional integration with Campaign Monitor. Ready for I18n.

- [front_matter_parser](#)

Ruby library to parse files or strings with YAML front matters. It has automatic syntax detection.

- [gutenweb](#)

A typographical CSS stylesheet to quick generation of documents with vertical rhythm and defined styles

- [CalibreRecipes](#)

Some calibre recipes

[@waiting-for-dev](#) on GitHub

Copyright © 2016 - Marc Busqué - Powered by [Octopress](#)