

Rails

This plugin adds completion for [Ruby On Rails Framework](#) and [Rake](#) commands, as well as some aliases for logs and environment variables.

To use it, add `rails` to the plugins array in your zshrc file:

```
plugins=(... rails)
```

List of Aliases

Rails aliases

Alias	Command	Description
rc	rails console	Interact with your Rails app from the CLI
rcs	rails console --sandbox	Test code in a sandbox, without changing any data
rd	rails destroy	Undo a generate operation
rdb	rails dbconsole	Interact with your db from the console
rdc	rails db:create	Create the database
rdd	rails db:drop	Delete the database
rdm	rails db:migrate	Run pending db migrations
rdmd	rails db:migrate:down	Undo specific db migration
rdmr	rails db:migrate:redo	Redo specific db migration
rdms	rails db:migrate:status	Show current db migration status
rdmtc	rails db:migrate db:test:clone	Run pending migrations and clone db into test database
rdmu	rails db:migrate:up	Run specific db migration
rdr	rails db:rollback	Roll back the last migration
rdrs	rails db:reset	Delete the database and set it up again
rds	rails db:seed	Seed the database
rdsl	rails db:schema:load	Load the database schema
rdtc	rails db:test:clone	Clone the database into the test database
rdtp	rails db:test:prepare	Duplicate the db schema into your test database
rgen	rails generate	Generate boilerplate code
rgm	rails generate migration	Generate a db migration
rlc	rails log:clear	Clear Rails logs

rm	rails middleware	Interact with Rails middlewares
rn	rails notes	Search for notes (FIXME, TODO) in code comments
rp	rails plugin	Run a Rails plugin command
rr	rails routes	List all defined routes
rrg	rails routes grep	List and filter the defined routes
rs	rails server	Launch a web server
rsb	rails server --bind	Launch a web server binding it to a specific IP
rsd	rails server --debugger	Launch a web server with debugger
rsp	rails server --port	Launch a web server and specify the listening port
rsts	rails stats	Print code statistics
rt	rails test	Run Rails tests
ru	rails runner	Run Ruby code in the context of Rails

Foreman

Alias	Command	Description
fmns	foreman start	Interact with your Rails app from the CLI

Utility aliases

Alias	Command	Description
devlog	tail -f log/development.log	Show and follow changes to the development log
prodlog	tail -f log/production.log	Show and follow changes to the production log
testlog	tail -f log/test.log	Show and follow changes to the test log

Environment settings

Alias	Command	Description
RED	RAILS_ENV=development	Sets RAILS_ENV to development
REP	RAILS_ENV=production	Sets RAILS_ENV to production
RET	RAILS_ENV=test	Sets RAILS_ENV to test

These are global aliases. Use in combination with a command or just run them separately. For example: `REP rake db:migrate` will migrate the production db.

Legacy

Rake aliases

The following commands are run `using rails` instead of `rake` since Rails v5, but are preserved under the prefix `rk` for backwards compatibility.

Alias	Command	Description
rkdc	rake db:create	Create the database
rkdd	rake db:drop	Delete the database
rkdm	rake db:migrate	Run pending db migrations
rkdms	rake db:migrate:status	Show current db migration status
rkdmto	rake db:migrate db:test:clone	Run pending migrations and clone db into test database
rkdr	rake db:rollback	Roll back the last migration
rkdrs	rake db:reset	Delete the database and set it up again
rkds	rake db:seed	Seed the database
rkdsl	rake db:schema:load	Load the database schema
rkdtc	rake db:test:clone	Clone the database into the test database
rkdtp	rake db:test:prepare	Duplicate the db schema into your test database
rklc	rake log:clear	Clear Rails logs
rkmd	rake middleware	Interact with Rails middlewares
rkn	rake notes	Search for notes (FIXME, TODO) in code comments
rksts	rake stats	Print code statistics
rkt	rake test	Run Rails tests

Other

Alias	Command
sc	ruby script/console
sd	ruby script/destroy
sd	ruby script/server --debugger
sg	ruby script/generate
sp	ruby script/plugin
sr	ruby script/runner
ssp	ruby script/spec
sstat	thin --stats "/thin/stats" start

- `remote_console <server> <directory>` : runs `ruby script/console production` on a remote server.